

STUDY OF RANDOM FIELD MODELS FOR TEXTURE

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of
MASTER OF TECHNOLOGY

By
POOLAKMOY BANDYOPADHYAY

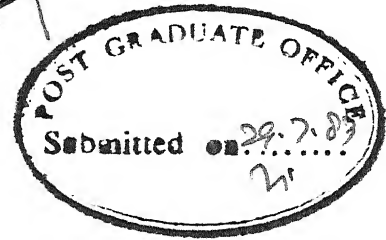
to the
DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY, KANPUR
JULY, 1983

13 JUN 1965

RECEIVED
AIR FORCE
87482

EE-1983-M-BAN-STU

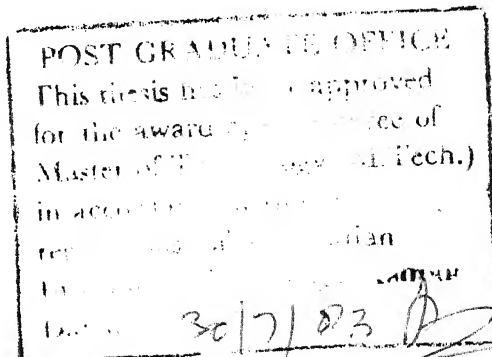
CERTIFICATE



Certified that this work 'STUDY OF RANDOM FIELD MODELS FOR TEXTURE' by POOLAKMOY BANDYOPADHYAY is carried out under my supervision and is not submitted elsewhere for a degree.



(S.K. Mullick)
Professor
Department of Electrical Engineering
Indian Institute of Technology
KANPUR



ACKNOWLEDGEMENT

I wish to take this opportunity to express my deepest sense of gratitude to my supervisor Dr. S.K. Mullick who has initiated me into this problem and provided me with infalliable guidance. His sincere advice, keen interest and constructive criticism during the course of this work has been a source of constant encouragement to me.

I don't have words to express my gratitude to Prof. R.L. Kashyap. Long discussions with him during his stay in IIT, helped me a lot to take up the challenges in the field of image processing.

I am also indebted to Prof. V.P. Sinha and Prof. J.N. Kapur for their constant encouragements and suggestions.

Thanks are due to Mr. K.N.H. Bhat, P.G. Poonacha and M.K. Patra for the fruitful discussions in various stages of the thesis.

I am also indebted to DRDO for financial support. I also thank all others who made my stay at IIT memorable and pleasant.

Finally I deeply appreciate the excellent typing done by Mr. J.S. Rawat.

P. Bandyopadhyay

LIST OF CONTENTS

| | Page |
|--|------|
| CHAPTER 1 INTRODUCTION | 1 |
| 1.1 Texture Models | 3 |
| 1.2 The problem and the approach | 4 |
| 1.3 Literature Survey | 6 |
| 1.4 Outline of the Thesis | 11 |
| CHAPTER 2 CAUSAL SPATIAL MODEL FOR TEXTURES | 13 |
| 2.1 Introduction | 13 |
| 2.2 Review of 1D time Series Model | 14 |
| 2.3 Principle of Parameter Estimation | 17 |
| 2.4 2D Statistical Model | 23 |
| 2.5 Empirical Equations for Model Building | 30 |
| 2.6 Model Building | 32 |
| 2.7 Simulation Results | 35 |
| CHAPTER 3 FREQUENCY DOMAIN CHARACTERIZATION OF STOCHASTIC TEXTURES | 42 |
| 3.1 Introduction | 42 |
| 3.2 PN Sequences and Arrays | 43 |
| 3.3 Pseudo-random Arrays: Generation | 48 |
| 3.4 Spatial Filter Model | 51 |
| 3.5 Texture Analysis | 52 |
| 3.6 Parameter Extraction | 53 |
| 3.7 Use of Separable Filter to Compute Phase | 60 |
| 3.8 Simulation Results | 62 |

| | Page |
|--|------|
| CHAPTER 4 ANALYSIS AND SYNTHESIS OF TEXTURES BY NONCAUSAL RF MODELS | 67 |
| 4.1 Introduction | 67 |
| 4.2 Notations | 68 |
| 4.3 Summary of Discrete Spatial Interaction Models | 70 |
| 4.4 Infinite Lattice Models | 72 |
| 4.5 Finite Lattice Models | 75 |
| 4.6 Analysis of Finite Lattice Models | 79 |
| 4.7 Properties of FL Models | 85 |
| 4.8 Relationship between SAR and CM Models | 87 |
| 4.9 Parameter Estimation | 89 |
| 4.10 Choice of Neighbourhood for SAR and CM Models | 93 |
| 4.11 Experimental Results | 97 |
| 4.12 Discussion | 106 |
| CHAPTER 5 TEXTURE SEGMENTATION BY SAR MODELS | 107 |
| 5.1 Principle of Segmentation | 107 |
| 5.2 Application of RF Model for Segmen- tation | 108 |
| 5.3 Principle of Clustering | 110 |
| 5.4 Simulation Results | 115 |
| CHAPTER 6 CONCLUSION | 124 |
| REFERENCES | 126 |

ABSTRACT

This study considers statistical modelling of a special type of images, called textures. Both causal and noncausal neighbourhoods are considered. We consider the 2D time series models for textures and the parameter estimation schemes for such models. We also discuss the frequency domain characterization of textures and application of PN arrays for this simulations. Among the non-causal models, we present two important ones; the simultaneous autoregressive (SAR) and conditional Markov (CM). Each of these models is characterized by a set of neighbours, a set of coefficients and a noise sequence of specified properties. To avoid excessive computation, we use an approximate maximum likelihood scheme for estimating the parameters. Asymptotically consistent decision rules are given for choosing an appropriate SAR or CM model. Model studies have been substantiated using synthetic textures. An image segmentation scheme based on SAR model is proposed and studied in detail. The programs generated for this study are also useful for images other than textures.

CHAPTER 1

INTRODUCTION

The subject of image modelling involves the construction of models or procedures for the specification of images. These models serve a dual purpose that they can describe images that are observed and also can serve to generate synthetic images from the model parameters. We will be concerned with the models of a specific type of images, the class of models for representing textures in image. Proper modelling of image has become important because, the efficient design of computer algorithms for image analysis, processing and synthesis can only be done using the framework of an image model.

There are four important areas of image processing in which texture plays an important role: classification [1,2], image segmentation [3], realism in computer graphics [4] and image encoding [5]. Besides being an intrinsic feature of realistic objects, texture also gives important information on the depth and orientation of an object [6]. Julesz [7] considers the problem of generation of familiar textures, an important one from both practical and theoretical viewpoints. Understanding texture is also an essential part of understanding of human vision [8]. These considerations

have led to an increased activity in the area of texture analysis and synthesis.

There is no universally accepted definition for textures. Part of the difficulty in giving a definition of texture is the extremely large number of attributes of texture that we would like to include in the definition. We consider a texture to be stochastic, possibly periodic, 2D image field. The most important attributes of texture are coarseness, contrast, directionality, line linkness, regularity and roughness.

Most texture researches concerning texture modelling are characterized by the underlying assumptions made about the texture formation process. There are two major assumptions, and the choice of these assumptions depends primarily on the type of texture to be considered in the study. The first assumption which is called the placement rule viewpoint, considers a texture to be composed of primitives. These primitives may be of random or of deterministic shape, such circles, polygons or even dot patterns. Macrotextures have large primitives, whereas microtextures are composed of small primitives. The texture image is formed from the primitives by placement rules which specify how the primitives are oriented, both on the image field and with respect to each other. Examples of such textures include tiling

of the plane, cellular structures and a picture of brick wall. The second viewpoint regarding texture generation processes involves the stochastic assumption. The placement rule principle for textures may include a random aspect in the stochastic point of view, however, we may consider that the texture is a sample from a probability distribution on the image space. The image space is usually a square lattice and the value at each grid point in the lattice is a random variable (non-negative). Textures such as sand, grass and water are not appropriately described by a placement rule model. The key feature of these images is that the primitives are random in shape and cannot be easily described.

1.1 TEXTURE MODELS

By the model of a texture, we mean a mathematical process which creates or describes the textured image. The primary object of texture modelling is the description of real textures. A secondary goal of texture modelling is classification of textures. The numerical parameters of the model can be used as features to classify the textures. Texture modelling is an important concept in the sense that it gives the inherent features of the image and hence helps further processing. A survey of the well known and commonly used textures model schemes has been given by Ahuja [9]. The

problem of characterization of stochastic texture is basically a system identification problem where the system is a two dimensional one.

1.2 THE PROBLEM AND THE APPROACH

In this thesis, we consider different approaches for characterizing a stochastic texture. Among the various methods of modelling stochastic texture, we only consider the random field (RF) models as they are mathematically tractable. In RF model of texture, we assume that the texture is generated by a linear system [10] when the system is excited by white noise. The different pixel values of the texture follow a probability distribution depending on the system and the probability distribution of the input. This problem of system identification can be tackled in two ways

- i) Spatial Domain Approach
- ii) Transform Domain Approach

In spatial domain approach, we represent the input-output relationship by a difference equation. Based on the dependency of any pixel on other pixels in a given region, we define a spatial model to be either causal or noncausal and they are shown in Fig. 1.1.

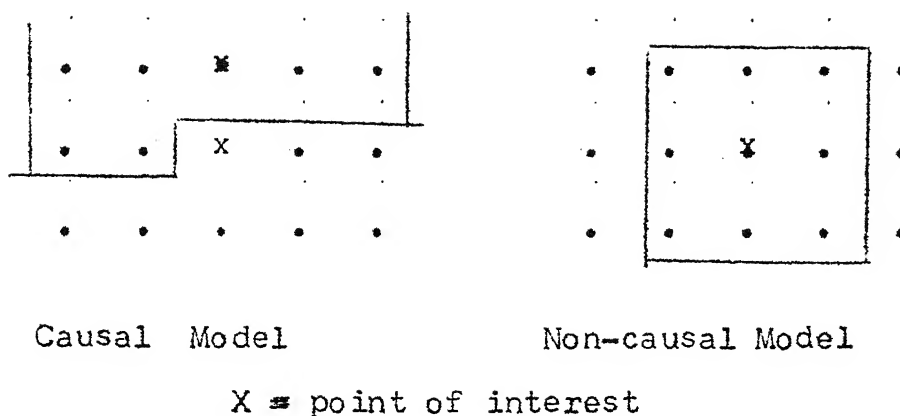


Fig. 1.1: Spatial Neighbourhood for images

In case of noncausal model, we assume that the value of a pixel depends on all the neighbouring pixel values in a given region. In spatial domain characterization, the problem is to estimate the coefficient or the parameters of the governing difference equation. Different statistical schemes are generally used to estimate the parameters. For noncausal model, we assume the image to be torus, finite lattice image and depending on the input sequence, these noncausal models are of two type

- i) Simultaneous Auto Regressive (SAR)
- ii) Conditional Markov (CM)

In SAR model, we assume that the input to the system is uncorrelated but it is correlated one in case of CM model. These types of noncausal models are very useful in texture analysis due to their realistic assumption about the neighbourhood.

In the transform domain characterization, texture is viewed as the output of a homogeneous spatial filter excited by white, not necessarily Gaussian noise. Different statistical measures give the amplitude and phase response of the filter. In identifying the system, this method is not as good as the parameter estimation scheme but this is a computationally efficient scheme due to fast transform algorithms.

1.3 LITERATURE SURVEY

The random field model of textures has been studied by many researchers in the recent years. Depending on the type of image, the random field model has two subclasses

- i) Global models
- ii) Local models

Global models treat an entire image as the realization of a random field. Hunt [11,12] has proposed a nonstationary gaussian RF model for textures. He has shown the appropriateness of this model by subtracting the local ensemble average from each image point and showing that the resulting image fits a stationary Gaussian model. In a recent paper, Hunt [13], discusses three kinds of nonstationarities:

- Case 1: Nonstationary mean, nonstationary autocorrelation.
- Case 2: Nonstationary mean, stationary autocorrelation.

Case 3: Stationary mean, nonstationary autocorrelation.

The nonstationarity in the image mean is described by the array of individual means taken over a specified neighbourhood about each image point. For the autocorrelation function, the breakdown of stationarity is related to the way the correlation function changes over the image. Hunt uses three attributes of the correlation function

to describe its spatial dependence: its energy, its width, and its shape. Trussel and Kruger [14] claim that the Laplacian density function is better suited for modelling high pass filtered imagery than the Gaussian function. They have also shown that the basic assumption which allows the Gaussian model to be used for image restoration purposes are still valid under a Laplacian model.

Nahi and Jananshahi [15] suggest modelling the texture by background and foreground statistical processes. They assume that the two portions are statistically independent random processes with known (or estimated) first two moments.

Recursive solutions based on differential equations are common in one-dimensional signal processing and have been generalized to two dimensions. Jain [16] investigates

the applicability of three kinds of random fields to the image modelling problem, each characterized by a different class of partial differential equations (PDEs): hyperbolic, parabolic and elliptic. A digital shape is defined by a finite difference approximation of a PDE. The class of hyperbolic PDE are shown to provide more general causal models than autoregressive moving average models. For a given spectral density function, parabolic PDEs can provide causal, semi-causal and noncausal representations. Elliptic PDEs provide noncausal models that represent two dimensional discrete Markov field and can be used to model both isotropic and anisotropic imagery. Jain argues that PDE models is based on a well-established mathematical theory and, further, that there exists a considerable body of computer software for numerical solutions. System identification techniques may be used for choosing the PDE model for a given class of image.

Pratt and Faugeras [17] and Gagalowicz [18] view texture as the output of a homogeneous spatial filter, excited by white, not necessarily Gaussian noise. A texture is characterised by its mean, the histogram of white noise and the transfer function of the filter. For a given texture, the model parameters are obtained as follows

- i) The mean is readily estimated from the texture.
- ii) The autocorrelation function is computed to determine the amplitude response of the transfer function.
- iii) Higher order moments are computed to determine the phase response of the filter.

Inverse filtering yields the white noise image and hence its histogram and probability distribution. The inverse filtering or decorrelation may be done by simple operators. For example, for a first order Markov field decorrelation may be done by Laplacian operator [19]. The whitened field estimate of the independent identically distributed noise process will identify the spatial process in terms of autocorrelation function.

If neither knowledge nor hypotheses about the type of random field are available, a class of local models is used. These models assume relationships among gray levels of pixels in small neighbourhoods. The modelling process consists of choosing the relationship and evaluating its parameters. Two basic categories of local models arise from the joint and conditional probability formulations of variations in a neighbourhood proposed by Whittle [20] and Bartlett [21]. Besag [22], however described a number of difficulties with these approaches. He has shown

that the constraints on the conditional probability structure are so severe that they actually control the particular models. Nevertheless, the conditional probability approach serves as the basis for a commonly used class of models, called Markov image models. Different choices for the set of a pixel's neighbours give rise to different Markov models. The specification of the probability distribution of a pixel's gray level, given the gray level of its neighbours, defines a strict sense Markov field representation. Two of the more interesting models of this class have been discussed by Besag [23] and they are called autobinary and autonormal schemes.

A close relative of the automodel is the simultaneous autoregressive model (SAR). Kashyap [24] notes that the symmetry requirements of the SAR are automatically fulfilled without the need to place prior restrictions on them.

Kashyap [25] uses circulant (block) matrices to define SAR models for an infinite array (array obtained by repeating a given finite image). He has also given the scheme for estimating the parameters of the model and decision rules for the choice of optimum neighbours. Kashyap also discusses all these schemes for a doubly periodic array defined over a finite lattice.

The local models discussed so far relate the gray level of a pixel to the gray level of its neighbours, using a conditional probability formulation. Through this approach has got considerable attention, its primary difficulty, however, is the high dimensionality of the joint probability densities, even for small neighbourhoods, making parameter estimation complex and cumbersome.

Rosenfeld and Troy [26] and Haralick [27] suggest the use of two dimensional spatial dependence of gray levels for fixed distances and angular operations. Haralick and numerous other investigators apply features derived from the co-occurrence matrix to various texture classification and discrimination problems.

The entire area of syntactic image model is not covered here as it is outside the scope of this thesis.

1.4 OUTLINE OF THE THESIS

After a brief introduction in Chapter 1, in Chapter 2 we have studied the 2D time series model of textures. Synthesis and parameter estimation schemes for different auto-regressive and moving averages processes have been discussed. A procedure for checking the adequacy of the estimated parameter is also given.

In chapter 3, we have studied the transform domain characterization of texture. Instead of computer generated

random number, simulation is carried out by using non-binary pseudo-random array, being ideally white in the cyclic sense. The amplitude and phase response have been estimated for a separable spatial filter.

Chapter 4 discusses noncausal models for a finite lattice image. These images are assumed to be toroidal. Both simultaneous autoregressive (SAR) and conditional Markov (CM) images have been studied. For parameter estimation, we have followed Kashyap's method for obtaining an approximate maximum likelihood estimate and a decision rule for optimum neighbourhood has also been studied. We have also discussed the spectral density characterization for such models.

In Chapter 5, we have proposed an application of spatial interaction model to texture segmentation. Simulations have been carried out for different segmenting curves and for all such cases the proposed scheme worked satisfactorily. Some of the disadvantages and error performance of this method have also been discussed. Segmentation scheme is studied only for SAR models.

Chapter 6 sums up the contribution of the thesis with recommendation for further work along the same line.

CHAPTER 2

CAUSAL SPATIAL MODEL FOR TEXTURES

2.1 INTRODUCTION

Time series modelling of any observed set of data is an old concept and is used extensively in statistics as well as engineering. These models are also being used successfully in social sciences and economics. Their successful application to 1D data has led researchers to apply such models to multidimensional data; in particular to 2D spatial processes. In the area of image processing such models are currently being used in image modelling and subsequently, using these models, to tackle problems of image restoration, texture classification, image data compression, spectral analysis etc. [9].

In recent years, however, considerable efforts have been made to model a 2D data array (e.g. images) by models which will take care of the dependency of a point on its spatial neighbourhood rather than on its 'past' as the concept of 'past and future' are not defined naturally in such cases.

In this chapter we will study two dimensional time series models and their parameters estimation, but this 2D analysis will be done only after a brief introduction to its one-dimensional counterpart.

2.2 REVIEW OF 1D TIME SERIES MODEL

Let a time series be represented by $y_1, y_2, \dots, y_i, \dots$ and let $\{\omega_i\}$ represent an independent identically distributed (IID) sequence of zero mean and variance σ_ω^2 . Then the time series may be modelled by

$$\begin{aligned} y_i &= \mu + \omega_i + \alpha_1 \omega_{i-1} + \alpha_2 \omega_{i-2} + \dots \\ &= \mu + \sum_{k=0}^{\infty} \alpha_k \omega_{i-k} \end{aligned} \quad (2.1)$$

with $\alpha_0 = 1$

or y_i may be related to ω_i by the eqn.

$$\begin{aligned} y_i &= \beta_1 y_{i-1} + \beta_2 y_{i-2} + \dots + \omega_i \\ &= \sum_{j=1}^{\infty} \beta_j y_{i-j} + \omega_i \end{aligned} \quad (2.2)$$

Here the eqn. (2.1) represents a moving average (MA) process of infinite order whereas eqn. (2.2) represents an autoregressive process of infinite order. From eqn. (2.1) we can always get a moving average process of finite order (say q) as

$$y_i = \sum_{k=0}^q \alpha_k \omega_{i-k} + \mu \quad (2.3)$$

Here μ represents the mean of the output process.

Again from eqn. (2.2) we can get an auto-regressive process of finite order (say p) as

$$y_i = \sum_{j=1}^p \beta_j y_{i-j} + \omega_i \quad (2.4)$$

Eqn. (2.3) and eqn. (2.4) will be denoted as MA(q) and AR(p) processes respectively. By combining these two, we get an auto-regressive-moving average process where

$$y_i = \mu + \sum_{j=1}^p \beta_j y_{i-j} + \sum_{k=0}^q \alpha_k \omega_{i-k} \quad (2.5)$$

Eqn. (2.5) is often written as ARMA(p, q) process.

Assuming a zero mean process, eqn. (2.5) gives

$$\begin{aligned} y_i + a_1 y_{i-1} + \dots + a_p y_{i-p} &= b_0 x_i + b_1 x_{i-1} \\ &+ \dots + b_q x_{i-q} \end{aligned} \quad (2.6)$$

Relationship (2.6) can be represented by an input-output LTI model given by Fig. 2.1.

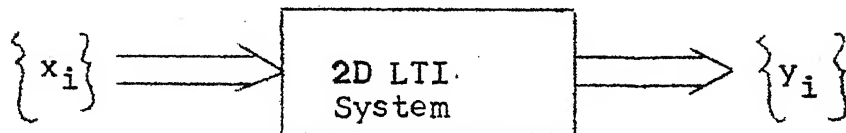


Fig. 2.1: A time series model of Texture.

Taking Z-transform on both sides of eqn. (2.6) we get,

$$(1+a_1z^{-1} + \dots + a_pz^{-p})Y(z)=(b_0+b_1z^{-1} + \dots + b_qz^{-q})X(z) \quad (2.7)$$

where the Z-transform is defined as

$$Y(z) = \sum_{i=-\infty}^{\infty} y_i z^{-i}$$

So the transfer function of the LTI system is

$$G(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + \sum_{k=1}^q b_k z^{-k}}{1 + \sum_{k=1}^p a_k z^{-k}} \quad (2.8)$$

for one sided Z-transform.

Case 1:

Let us assume $b_k = 0$ for $k = 1, 2, \dots, q$. Eqn. (2.8) yields

$$G(z) = \frac{b_0}{1 + \sum_{k=1}^p a_k z^{-k}} \quad (2.9)$$

By taking inverse Z-transform

$$(1 + \sum_{k=1}^p a_k z^{-k}) Y(z) = b_0 X(z) \quad (2.10)$$

and

$$y_i = \sum_{k=1}^p (a_k) y_{i-k} + b_0 x_i \quad (2.11)$$

which is an auto-regressive process of order p . So it is seen that in case of an auto-regressive process, the system is an all-pole system.

Case 2:

If we assume $a_k = 0$ for $k = 1, 2, \dots, p$, equation (2.8) reduces to

$$G(z) = b_0 + \sum_{k=1}^q b_k z^{-k} \quad (2.12)$$

From eqn. (2.12) we get

$$Y(z) = b_0 + \sum_{k=1}^q b_k z^{-k} \quad (2.13)$$

and

$$y_i = \sum_{k=0}^q b_k x_{i-k} \quad (2.14)$$

This, in comparison to eqn. (2.3) is a moving average process of order q .

Again it is seen from eqn. (2.13) that, in case of moving average process, the system is an all zero system. From eqn. (2.8) we see that the system in case of ARMA(p, q) process is a pole-zero system. In this respect ARMA comes out to be most general one.

2.3 PRINCIPLE OF PARAMETERS ESTIMATION

The auto-covariance for the case of AR(p) process is,

$$\gamma_k = E[y_i y_{i-k}] \quad (2.15)$$

where $E[.]$ represents statistical expectation. Then from eqn. (2.4) we get, by taking expectation on both sides

$$\gamma_k = \beta_1 \gamma_{k-1} + \beta_2 \gamma_{k-2} + \dots + \beta_p \gamma_{k-p} \quad (2.16)$$

The autocorrelation coefficient is defined as

$$\rho_k = \gamma_k / \gamma_0 \quad (2.17)$$

Hence from (2.16) and (2.17) we get,

$$\rho_k = \beta_1 \rho_{k-1} + \beta_2 \rho_{k-2} + \dots + \beta_p \rho_{k-p} \quad (2.18)$$

For $k = 1, 2, 3, \dots, p$, from eqn. (2.18), we get a set of linear equations

$$\left. \begin{aligned} \rho_1 &= \beta_1 \rho_0 + \beta_2 \rho_0 + \dots + \beta_p \rho_0 \\ \rho_2 &= \beta_1 \rho_1 + \beta_2 \rho_0 + \dots + \beta_p \rho_{p-2} \\ &\vdots \\ \rho_p &= \beta_1 \rho_{p-1} + \beta_2 \rho_{p-2} + \dots + \beta_p \rho_0 \end{aligned} \right\} \quad (2.19)$$

This is the well known Yule-Walker [28] equations for AR processes. We can always find a matrix-vector relationship for eqn. (2.19). There is a fast algorithm developed by Levinson to solve such a set of linear equation for estimation the parameters β_i [29].

If we assume ϕ_{ki} be the i th coefficient in a k th AR process, then Yule Walker equation becomes

$$\begin{bmatrix} 1 & p_1 & p_2 & \cdots & p_{k-1} \\ p_1 & 1 & p_1 & \cdots & p_{k-2} \\ \vdots & \vdots & \vdots & & \vdots \\ p_{k-1} & p_{k-2} & p_{k-3} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \phi_{k1} \\ \phi_{k2} \\ \vdots \\ \phi_{kk} \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_k \end{bmatrix} \quad (2.20)$$

The left hand side matrix in eqn. (2.20) is of Toeplitz nature, so any fast method can also be used for solving such equations. In the set of coefficients $\{\phi_{ki}\}_{i=1, \dots, k}$, the last one, i.e., ϕ_{kk} is of great importance and it is called partial autocorrelation, and in an AR process ϕ_{kk} gives an estimate of the order of the process. From eqn. (2.20) we get,

$$\phi_{11} = p_1$$

$$\phi_{22} = \frac{\begin{vmatrix} 1 & p_1 \\ p_1 & 1 \end{vmatrix}}{\begin{vmatrix} 1 & p_1 \\ p_1 & 1 \end{vmatrix}}$$

$$\phi_{33} = \frac{\begin{vmatrix} 1 & p_1 & p_2 \\ p_1 & 1 & p_2 \\ p_2 & p_1 & 1 \end{vmatrix}}{\begin{vmatrix} 1 & p_1 & p_2 \\ p_1 & 1 & p_1 \\ p_2 & p_1 & 1 \end{vmatrix}} \text{ etc.}$$

In fact, partial autocorrelation function of any process reveals an interesting property of the model. It can be shown that PAC (partial autocorrelation) function ϕ_{kk} for AR(p) process follows

$$\phi_{kk} \neq 0 \quad \text{for } k \leq p$$

$$\phi_{kk} = 0 \quad \text{for } k > p$$

Hence the PAC of a pth order AR process has a cutoff after lag p.

The moving average process, in comparison with AR process, is more complicated to analyse. Let us take the MA(q) process of eqn. (2.3) for the sake of analysis

$$\bar{y}_i = y_i - \mu$$

$$\bar{y}_i = \omega_i + \alpha_1 \omega_{i-1} + \dots + \alpha_q \omega_{i-q} \quad (2.21)$$

Again, for such MA(q) process, the covariance is defined as

$$\gamma_k = E [\bar{y}_i \bar{y}_{i-k}]$$

$$\begin{aligned} \therefore \gamma_0 &= E [\bar{y}_i \bar{y}_i] \\ &= (1 + \alpha_1^2 + \alpha_2^2 + \dots + \alpha_q^2) \omega^2 \\ &= D_q \omega^2 \end{aligned} \quad (2.22)$$

where $D_q = (1 + \alpha_1^2 + \alpha_2^2 + \dots + \alpha_q^2)$

and $\sigma_\omega^2 = E[\omega_i \omega_i] = \text{input variance} \quad (2.23)$

Since we have assumed that the input sequence is zero mean and IID, it follows that

$$E[\omega_i] = 0$$

$$\text{and } E[\omega_i \omega_j] = 0 \text{ for } i \neq j$$

From eqn. (2.21) it follows that autocovariance γ_k at lag k is given by

$$\begin{aligned} \gamma_k &= (\alpha_k + \alpha_1 \alpha_{k+1} + \dots + \alpha_{q-k} \alpha_q) \sigma_\omega^2 \\ &\quad \text{for } k = 1, 2, \dots, q \\ &= 0 \quad \text{for } k > q \end{aligned} \quad (2.24)$$

So combining eqn. (2.22) and eqn. (2.24), we get, auto-correlation

$$\begin{aligned} \rho_k &= \frac{\gamma_k}{\gamma_0} = \frac{\alpha_k + \alpha_1 \alpha_{k+1} + \dots + \alpha_{q-k} \alpha_q}{D_q} \quad \text{for } k=1, 2, \dots, q \\ &= 0 \quad \text{for } k > q \end{aligned} \quad (2.25)$$

From eqn. (2.25), for $k = 1, 2, 3, \dots$, we get a set of non-linear equations as follows

$$\left. \begin{aligned} \rho_1 D_q &= \alpha_1 + \alpha_1 \alpha_2 + \dots + \alpha_{q-1} \alpha_q \\ \rho_2 D_q &= \alpha_2 + \alpha_1 \alpha_3 + \dots + \alpha_{q-2} \alpha_q \\ &\vdots \\ \rho_q D_q &= 2\alpha_q \end{aligned} \right\} \quad (2.26)$$

These equations can be solved for getting the estimates of the parameters. It is evident that, for a lower order MA process, the solution may be obtainable but when order of MA process becomes large, the solution becomes less tractable.

An autoregressive moving average (ARMA) process is much more complicated than both AR and MA process in the sense that it contains two sets of parameters $\{\alpha_i\}$ and $\{\beta_i\}$. In ARMA process, the parameter estimation is again done after computing the correlation coefficients. In case of ARMA(p,q) process, these will be q autocorrelations f_q, f_{q-1}, \dots, f_1 whose values depend directly on the choice of q moving average parameters α as well as p autoregressive parameters β . Also the p values $f_q, f_{q-1}, \dots, f_{q-p+1}$ provide the necessary starting values for the difference equation $\phi(B) \quad k = 0 \quad \forall k \geq q+1$, which entirely determines the autocorrelation at higher lags. If $q-p < 0$, the whole autocorrelation function $f_j \quad \forall j = 0, 1, 2, \dots$, will consist of a mixture of damped exponentials and/or damped sine waves, whose nature is governed by the polynomial $\phi(B)$ and the starting values. If however $q-p \geq 0$, there will be a $q-p+1$ initial values f_0, f_1, \dots, f_{q-p} which do not follow this pattern. These facts are useful in identifying mixed processes.

Since parameter estimation in ARMA processes is not a simple procedure, Box and Jenkins have given a chart (P-520) which will give the initial estimate of the parameters for a ARMA(1,1) process.

2.4 2-D STATISTICAL MODEL [30]

In the previous section we have studied the analysis of different time series models, viz., AR, MA and ARMA. Those methods can be applied to 2D discrete images only when the image array is converted to a 1D vector by some scheme (like lexicographic ordering). If z_{ij} is representing a particular pixel value in the (i,j) position of the lattice then, it can be expressed again as a weighted sum of the data and white noise sequence (IID) within a particular region or neighbourhood. A portion of such a lattice is shown in Fig. 2.2, and all, the grid points are equally spaced.

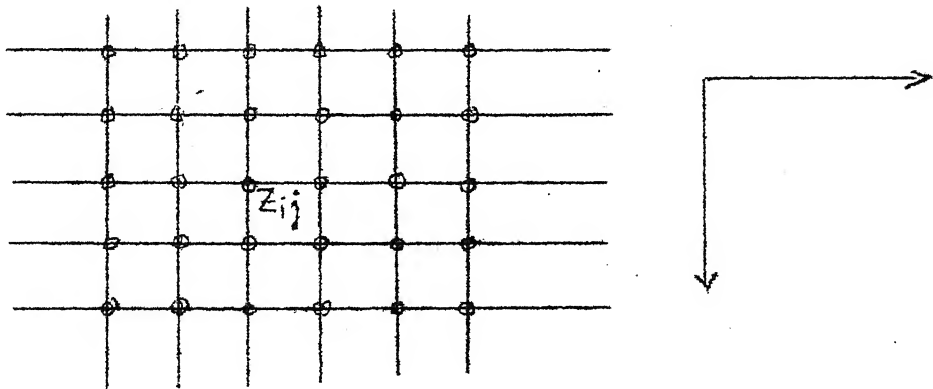


Fig. 2.2: 2D Image Lattice.

Hence in 2 dimension, the moving average process is given by

$$z_{i,j} = \omega_{i,j} + 0,1 \omega_{i,j} + 1,0 \omega_{i-1,j} + 1,1 \omega_{i-1,j} + \dots \quad (2.27)$$

If we define the horizontal shift operator by H , then,

$$H^n z_{i,j} = z_{i,j-n} \quad (2.28)$$

Similarly a vertical shift operator is defined by V where

$$V^m z_{i,j} = z_{i-m,j} \quad (2.29)$$

Two more parameters need to be defined for the sake of analysis they are horizontal difference operator and vertical difference operator defined as follows.

Horizontal difference Operator:

$$\begin{aligned} \nabla_h &= 1 - H \\ \nabla_h z_{i,j} &= (1-H)z_{i,j} = z_{i,j} - z_{i,j-1} \end{aligned} \quad (2.30)$$

and vertical difference operator:

$$\begin{aligned} \nabla_v &= 1 - V \\ \nabla_v z_{i,j} &= (1-V)z_{i,j} = z_{i,j} - z_{i-1,j} \end{aligned} \quad (2.31)$$

These operators are commutative

$$H^m V^n z_{i,j} = V^n H^m z_{i,j}$$

and

$$\nabla_n^m \nabla_v^n z_{i,j} = \nabla_v^n \nabla_h^m z_{i,j}$$

If we define

$$\Psi(H,V) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \varphi_{n,m} H^m V^n \quad (2.32)$$

then eqn. (2.27) reduces to

$$z_{i,j} = \Psi(H,V) \omega_{i,j} \text{ with } \varphi_{0,0} = 1 \quad (2.33)$$

Since $\Psi(H,V)$ represents the system whose output is the 2D time series $\{z_{i,j}\}$ hence it may be called the system function of the model.

Again in two-dimension, the auto-covariance at a lag (k, ℓ) is defined as

$$\gamma_{k,\ell} = E[z_{i,j} z_{i+k,j+\ell}] \quad (2.34)$$

From eqn. (2.33) we get,

$$\begin{aligned} \gamma_{k,\ell} &= E[\Psi(H,V) \omega_{i,j} \Psi(H,V) \omega_{i+k,j+\ell}] \\ &= \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \varphi_{p,q} \varphi_{m,n} \\ &\quad E[\omega_{i-p,j-q} \omega_{i+k-m,j+\ell-n}] \end{aligned}$$

This expression is nonzero only when $p=m-k$ and $q=n-l$ and for that value

$$E[\omega_{i-p,j-q}, \omega_{i-p,j-q}] = \sigma_{\omega}^2$$

$$\therefore \gamma_{k,l} = \sigma_{\omega}^2 \sum_{p=0}^{\infty} \sum_{q=0}^{\infty} \varphi_{p,q} \varphi_{p+k,q+l} \quad (2.35)$$

The autocovariance coefficient $\gamma_{k,l}$ at lag (k,l) can also be derived from the model system function $\underline{\Psi}(H,V)$. Let us define a function $\Gamma(H,V)$

$$\Gamma(H,V) \triangleq \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \gamma_{k,l} H^l V^k \quad (2.36)$$

Then from eqn. (2.35) we get,

$$\begin{aligned} \Gamma(H,V) &= \sigma_{\omega}^2 \sum \sum \sum \sum \varphi_{p,q} \varphi_{p+k,q+l} V^k H^l \\ &= \sigma_{\omega}^2 \sum \sum \sum \sum \varphi_{p,q} \varphi_{m,n} V^{m-p} H^{n-q} \\ &= \sigma_{\omega}^2 \left(\sum \sum \varphi_{m,n} V^m H^n \right) \left(\sum \sum \varphi_{p,q} V^{-p} H^{-q} \right) \\ &= \sigma_{\omega}^2 \underline{\Psi}(H,V) \underline{\Psi}(H^{-1}, V^{-1}) \end{aligned} \quad (2.37)$$

Again for the sake of analytical simplicity we study the estimation technique for 2D AR and MA process.

Case 1:

AR(m,n) process:

$$\text{Here } \Phi(H,V) z_{i,j} = \omega_{i,j} \quad (2.38)$$

$$\text{where } \Phi(H,V) = \sum_{k=0}^m \sum_{\ell=0}^n \phi_{k,\ell} V^k H^\ell$$

and $\phi_{0,0} = 1$

Case 2:

MA(p,q) process:

$$\text{Here } z_{i,j} = \Theta(H,V) \omega_{i,j} \quad (2.39)$$

$$\text{where } \Theta(H,V) = \sum_{k=0}^p \sum_{\ell=0}^q \theta_{k,\ell} V^k H^\ell$$

Case 3:

ARMA (m,m; p,q) process:

$$\text{Here } \Phi(H,V) z_{i,j} = \Theta(H,V) \omega_{i,j} \quad (2.40)$$

Like one dimensional ARMA process, the two dimensional ARMA process is complicated to analyse and the estimation of parameters in this case is not a straight forward procedure. In the remainder of this chapter we will study the AR and MA process in two dimension and will give an empirical rule to estimate the parameters. Test

for stationarity is also given for a given discrete data set.

Let us consider an AR(1,1) and MA(1,1) model which are the simplest non-trivial 2-D processes.

Case 1:

AR(1,1) Process:

$$\Phi(H,V) z_{i,j} = \omega_{i,j}$$

$$\text{Here, } \Phi(H,V) = 1 + \phi_{1,0} V + \phi_{0,1} H + \phi_{1,1} H V$$

The auto-covariance is

$$\gamma_{k,l} = \phi_{1,0} \gamma_{k-1,l} + \phi_{0,1} \gamma_{k,l-1} + \phi_{1,1} \gamma_{k-1,l-1}$$

similarly

$$\rho_{k,l} = \phi_{1,0} \rho_{k-1,l} + \phi_{0,1} \rho_{k,l-1} + \phi_{1,1} \rho_{k-1,l-1}$$

Since we are considering AR(1,1) process, hence by putting pairwise value of (k,l) as (0,1), (1,0) and (1,1) we get the corresponding 2D Yule-Walker equation as

$$\begin{bmatrix} 1 & \rho_{1,1} & \rho_{01} \\ \rho_{11} & 1 & \rho_{10} \\ \rho_{01} & \rho_{10} & 1 \end{bmatrix} \begin{bmatrix} \phi_{10} \\ \phi_{01} \\ \phi_{11} \end{bmatrix} = \begin{bmatrix} \rho_{10} \\ \rho_{01} \\ \rho_{11} \end{bmatrix}$$

The two dimensional correlation coefficient at lag (k, l) is defined as

$$\rho_{k,l} = \gamma_{k,l} / \gamma_{0,0}$$

The given matrix vector equation may be solved for the initial estimate of the parameter set. It is evident that dimension of the matrix and vectors depend on the order of process.

Case 2:

MA(1,1) process:

$$\text{Here } z_{ij} = \Theta(H, V) \omega_{ij}$$

$$\Theta(H, V) = 1 + \theta_{10}V + \theta_{01}H + \theta_{11}HV$$

Since the autocovariance of z_{ij} at lag (k, l) is given by

$$\gamma_{k,l} = E[z_{ij} z_{i-k, j-l}],$$

from the above equation we get

$$\gamma_{00} = (1 + \theta_{10}^2 + \theta_{01}^2 + \theta_{11}^2) \sigma_w^2$$

$$\gamma_{10} = (\theta_{10} + \theta_{11}\theta_{01}) \sigma_w^2$$

$$\gamma_{01} = (\theta_{01} + \theta_{11}\theta_{10}) \sigma_w^2$$

$$\gamma_{11} = \theta_{11} \sigma_w^2$$

So autocorrelation functions are

$$\rho_{10} = \frac{e_{10} + e_{11}e_{01}}{1 + e_{10}^2 + e_{01}^2 + e_{11}^2}$$

$$\rho_{01} = \frac{e_{01} + e_{11}e_{10}}{1 + e_{10}^2 + e_{01}^2 + e_{11}^2}$$

$$\rho_{11} = \frac{e_{11}}{1 + e_{10}^2 + e_{01}^2 + e_{11}^2}$$

From these equations we can solve for the parameters e_{10} , e_{01} and e_{11} . One thing to be noticed here is that the complexity of the parameter estimations increases with the order of the model, but in case of AR process if the order of the model is increased then the method for solving them remain unchanged in the sense that we have to solve more number of linear equations and same Yule Walker equation can be applied. So given a set of discrete data whether it is one dimensional or two dimensional, it is a general trend in all fields (where time series model works) to search for a suitable AR model.

2.5 EMPIRICAL EQUATIONS FOR MODEL BUILDING

In application of these methods, to a given image data, the ensemble averages will be replaced by sample averages for computational conveniences.

For an image $\{z_{ij}\}$ where its dimensions is $n_1 \times n_2$, we define:

1) Mean

$$\bar{z} = \frac{1}{n_1 n_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} z_{ij}$$

2) Autocovariance

$$\gamma_{k,\ell} = \frac{1}{(n_1-k)(n_2-\ell)} \sum_{i=1}^{n_1-k} \sum_{j=1}^{n_2-\ell} (z_{ij} - \bar{z})(z_{i+k,j+\ell} - \bar{z})$$

where (k,ℓ) is the lag in (i,j) direction.

3) Variance

$$\sigma_z^2 = \gamma_{00}$$

4) Autocorrelation $\rho_{k,\ell} = \frac{\gamma_{k,\ell}}{\gamma_{0,0}}$

5) Partial Autocorrelation function

a) In horizontal direction

$$\phi_{kk} = \rho_{0,1} \quad \text{for } k = 1$$

$$= \frac{\rho_{0,k} - \sum_{j=1}^{k-1} \phi_{(k-1),j} \rho_{0,(k-j)}}{1 - \sum_{j=1}^{k-1} \phi_{(k-1),j} \rho_{0,j}}, \quad \forall k = 2, 3, \dots$$

b) In vertical direction

$$\phi_{kk} = \rho_{0,1} \quad k = 1$$

$$= \frac{\rho_{k,0} - \sum_{j=1}^{k-1} \phi_{(k-1),j} \rho_{(k-j),0}}{1 - \sum_{j=1}^{k-1} \phi_{(k-1),j} \rho_{j,0}} \quad k = 2, 3, \dots$$

where

$$\phi_{(k-1),j} = \phi_{(k-2),j} - \phi_{(k-1),(k-1)} \phi_{(k-2),(k-j-1)}$$

$$\forall j = 1, 2, \dots, (k-1)$$

2.6 MODEL BUILDING

Given a set of 2D data we can not say by looking at it what type of model will be best for it. The information about the probable model (whether AR, MA or ARMA) can be obtained from the partial autocorrelation function. Box and Jenkins have shown [28] that the partial autocorrelation function gives an idea about the nature of the model as well as its probable order. They have given a chart of the nature of partial autocorrelation which is given here for AR, MA and ARMA.

We have also discussed in the previous section that a data set can be modelled as a time series model if and only

if it is a stationary one. A data set is said to be stationary if its covariance matrix is positive definite. The simple way to check stationarity of a series is by the behaviour of its autocorrelation function. In a stationary sequence the autocorrelation decays uniformly without any notch. In case of 2D data we have to test the smoothness of the autocorrelation in horizontal and vertical directions. With these things in mind we can give successive steps for model building for a given 2D data.

Step 1: Calculate the mean value

Step 2: Calculate the autocovariances

Step 3: Calculate the variance of the data sample

Step 4: Calculate the autocorrelations

Step 5: Check stationarity of the given data. A given data may not be stationary in vertical or horizontal directions.

This behaviour will be obvious from the autocorrelations.

In most of the cases, fortunately, it has been found that the differences in the adjacent pixel values in the nonstationary process remove the trend and make the process stationary. So, in an attempt (in general) to remove nonstationarity, we take differences in horizontal and vertical direction of the given 2D data till a stationary data series is apparent.

Step 6: Take the difference in the direction in which the data is nonstationary. The information about the direction of non-stationarity can be obtained from the partial autocorrelation function.

Step 7: Determine the partial ACF in order to check the approximate order of the model and check stationarity.

Step 8: Identify the model from Table 2.1.

Step 9: Estimate the parameters. After identifying a tentative model we can use the sample autocorrelations to obtain a preliminary estimate of the parameters. These values may be used as an initial estimate for obtaining maximum likelihood estimate.

For computing initial estimate of parameters in case of AR process, we have to solve a set of linear equations given by Yule Walker equation. But in case of MA parameter estimation, we have to solve a set of nonlinear equations and in that case determination of preliminary estimates is less tractable.

After identifying a tentative set of model parameters, for a given pictorial data, we will obtain the best-estimate of the parameters. This can be done by checking the randomness of the residual. The concept of residual in time

series model is as follows. Suppose, for any process (whether AR or MA), the actual value of parameter vector is $\underline{\theta}$. Given a IID sequence (preferably Gaussian), we can always generate an image \underline{y} by using the IID sequence and vector $\underline{\theta}$. Let us say $\hat{\underline{\theta}}$ is the estimate of the parameters. With the same IID sequence and parameter $\hat{\underline{\theta}}$ if we generate an image $\hat{\underline{y}}$ then the difference between \underline{y} and $\hat{\underline{y}}$ is called the residual. If the model is best one, then the residual y_r ($y_r = \underline{y} - \hat{\underline{y}}$) will be white noise sequence. This is what will be achieved in step 10.

Step 10: Compute the residuals and compute its autocorrelation functions. The autocorrelation of the residual will suggest the direction in which the model should be modified.

Step 11: Make modifications in the model based on the obtained information.

All these steps are also given as a flow chart in Fig.2.3.

2.7 SIMULATION RESULTS:

For simulation purposes, the textures were first generated using specific models and then the parameters were estimated from such synthetic textures.

For this purpose, we have only considered lower order AR and MA processes. Programs are also developed for generation of synthetic textures and subsequent parameter

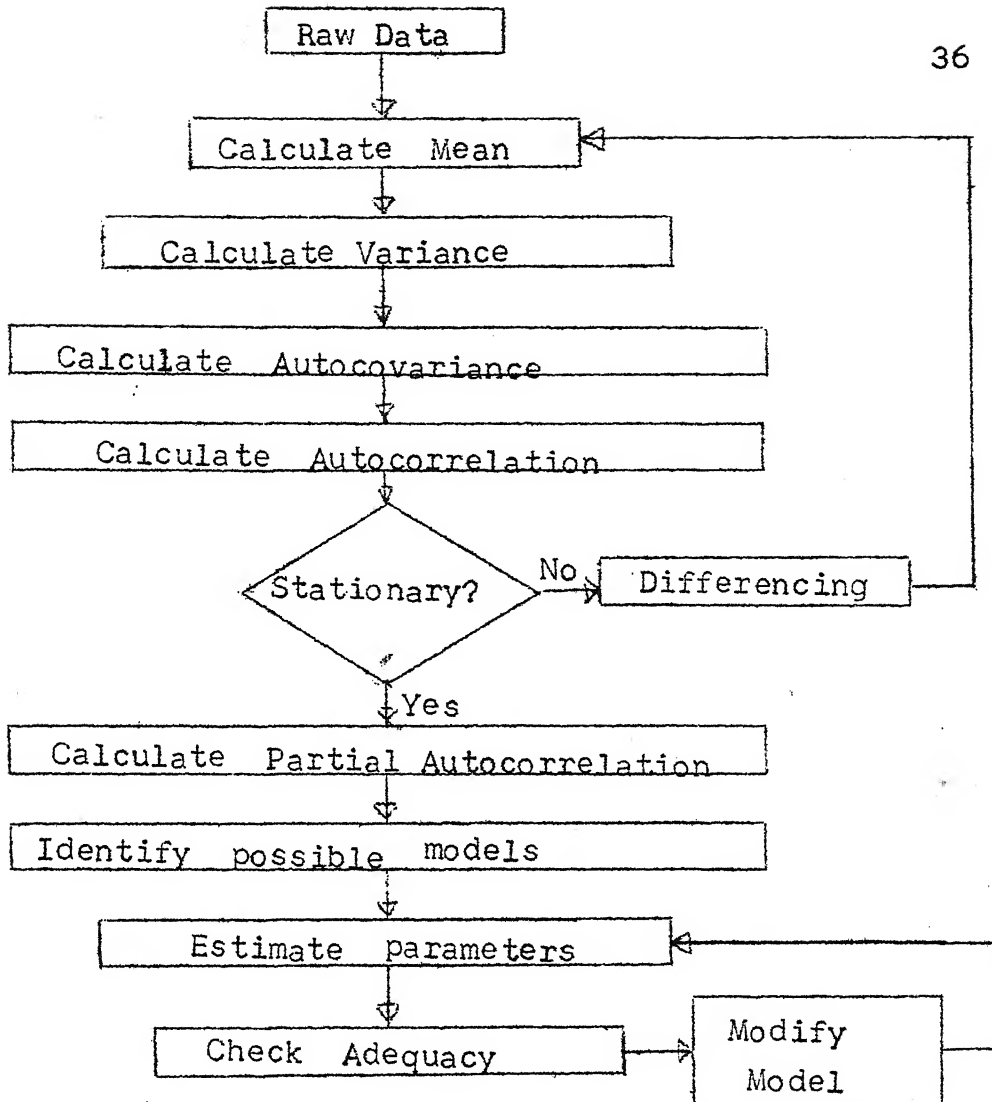
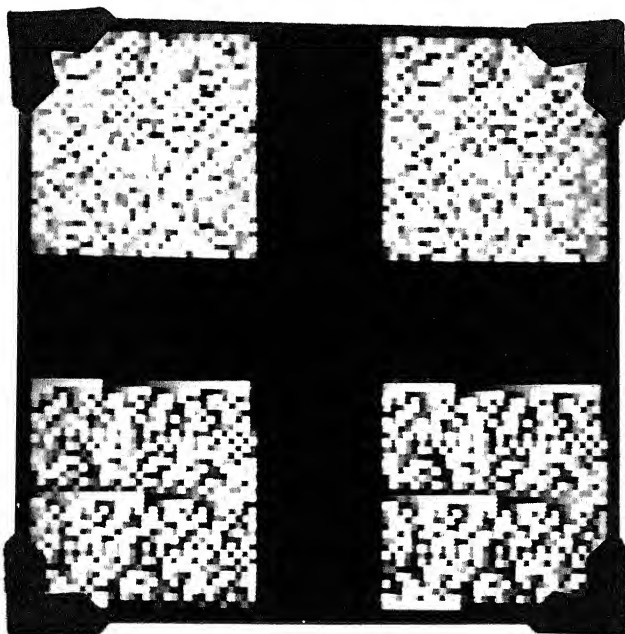


Fig. 2.2: Model Building Flow-chart

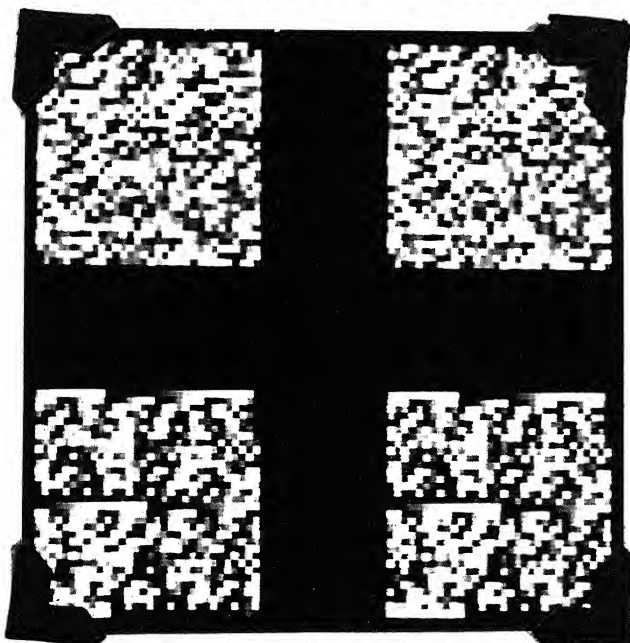
estimation and they are given in Appendix I. Provisions for handling higher order models is incorporated only for AR processes because the nonlinear nature of the autocorrelation equations in MA process cannot be predicted from the knowledge of lower order MA models.

In this study of 2D time series models, we have generated several synthetic images of size 64×64 and 32×32 . Some of such synthetic textures are shown in Fig. 2.4. In these cases we have used only $AR(1,1)$ and $MA(1,1)$ processes. Gaussian textures were only considered and for synthesis we have used IMSL (Intn'l Mathematical Subroutine Library) routine GGNOR for generating Gaussian IID sequence. But for 2D time series models we need 2D, IID sequence or an array. Since the elements of the sequence are independent so any arbitrary arrangement of them does not change their independence. Thus one way of generating a $M \times M$ IID array is to generate a $M^2 \times 1$ IID vector and then arrange them (lexicographically or by any other means) to form a $M \times M$ array. But considering the space requirements, we took another approach. In this method, each row (or column) of such IID array, was generated by using different seed value of GGNOR routine. From here onwards whenever such computer generated random number array with prescribed density is generated we have adopted the same scheme. Though we have mentioned in the

last section that for best estimate of parameters we have to take the maximum likelihood estimate, but it has been found that for image size of 64×64 the initial estimates are quite close to the actual parameter values which is explainable from statistical point of view. Instead of computing the maximum likelihood estimate which is a slow process in computers we have checked the whiteness of the residuals. Here we will give in a tabular form the actual and estimated parameter set for AR(1,1) and MA(1,1) processes for a 64×64 image. We have also given the plot of the autocorrelation function of the residuals for finite number of points for AR(1,1) and MA(1,1) processes, in Figs. 2.5 and 2.6.



2.4a: (1,1) original; (1,2) estimated (Gaussian AR(1,1))
 (2,1) original; (2,2) estimated (uniform AR(1,1))



2.4b: (1,1) original; (1,2) estimated (Gaussian MA(1,1))
 (2,1) original; (2,2) estimated (uniform MA(1,1))

Table 2.1: Summary of the Properties of AR, MA and ARMA Process [30]

| | AR Processes | MA Processes | ARMA Processes |
|------------------------------------|---|---|--|
| Model in terms of previous outputs | $\phi(B)z(n) = a(n)$ | $\phi^{-1}(B)z(n) = a(n)$ | $\phi^{-1}(B)\phi(B)z(n) = a(n)$ |
| Model in terms of previous inputs | $z(n) = \phi^{-1}(B)a(n)$ | $z(n) = \theta(B)a(n)$ | $z(n) = \phi^{-1}(B)\theta(B)a(n)$ |
| Wts. with outputs | Finite series | infinite series | infinite series |
| Wts. with inputs | infinite series | finite series | infinite series |
| Stationarity condition | roots of $\phi(B) = 0$ lie outside the unit circle | always stationary | roots of $\phi(B)$ lie outside the unit circle |
| Invertibility condition | always invertible | roots of $\theta(B)$ lie outside unit circle | roots of $\theta(B)$ lie outside unit circle. |
| Autocorrelation function | infinite (damped exponentials and/or damped sine waves) | finite | infinite (damped exponentials and/or damped sine waves after first q - p lags for ARMA(p, q) process). |
| | tails off | cuts off | tails off |
| Partial autocorrelation function | finite | infinite (dominated by damped exponentials and/or sine waves) | infinite (dominated by damped exponentials and sine waves after first p - q lags). |
| | cuts off | tails off | tails off |

Table 2.2

IMAGE SIZE : 64 x 64

MODEL : AR(1,1)

Input : GAUSSIAN (0,1)

| Actual Parameter Values | Estimated Autocorrelations | Estimated parameter Values |
|-------------------------|----------------------------|----------------------------|
| $\phi_{10} = 0.40$ | $00 = 1.000$ | $\phi_{10} = 0.3601$ |
| $\phi_{01} = 0.40$ | $01 = 0.8551$ | $\phi_{01} = 0.4612$ |
| $\phi_{11} = 0.40$ | $10 = 0.9299$ | $\phi_{11} = 0.3864$ |
| | $11 = 0.8214$ | |
| $\phi_{10} = 0.21$ | $00 = 1.0000$ | $\phi_{10} = 0.1832$ |
| $\phi_{01} = -0.14$ | $01 = 0.7216$ | $\phi_{01} = -0.1216$ |
| $\phi_{11} = 0.12$ | $10 = 0.8126$ | $\phi_{11} = 0.1522$ |
| | $11 = 0.6924$ | |
| $\phi_{10} = 0.28$ | $00 = 1.0000$ | $\phi_{10} = 0.2612$ |
| $\phi_{01} = -0.12$ | $01 = 0.8216$ | $\phi_{01} = -0.1013$ |
| $\phi_{11} = 0.14$ | $10 = 0.9021$ | $\phi_{11} = 0.1567$ |
| | $11 = 0.7964$ | |

IMAGE SIZE : 64 x 64

MODEL : MA(1,1)

INPUT : GAUSSIAN (0,1)

| Actual Parameter Values | Estimated Auto-correlations | Estimated Parameter Values |
|-------------------------|-----------------------------|----------------------------|
| $\theta_{10} = 0.40$ | $\phi_{00} = 1.0000$ | $\theta_{10} = 0.3521$ |
| $\theta_{01} = 0.40$ | $\phi_{10} = 0.34698$ | $\theta_{01} = 0.3724$ |
| $\theta_{11} = 0.40$ | $\phi_{01} = 0.34698$ | $\theta_{11} = 0.3523$ |
| | $\phi_{11} = 0.25327$ | |
| $\theta_{10} = 0.25$ | $\phi_{00} = 1.0000$ | $\theta_{10} = 0.2723$ |
| $\theta_{01} = 0.12$ | $\phi_{10} = 0.2634$ | $\theta_{01} = 0.1412$ |
| $\theta_{11} = 0.14$ | $\phi_{01} = 0.1654$ | $\theta_{11} = 0.1623$ |
| | $\phi_{11} = 0.1448$ | |
| $\theta_{10} = 0.12$ | $\phi_{00} = 1.0000$ | $\theta_{10} = 0.1622$ |
| $\theta_{01} = 0.14$ | $\phi_{10} = 0.1686$ | $\theta_{01} = 0.1623$ |
| $\theta_{11} = 0.20$ | $\phi_{01} = 0.1176$ | $\theta_{11} = 0.1721$ |
| | $\phi_{11} = 0.1613$ | |

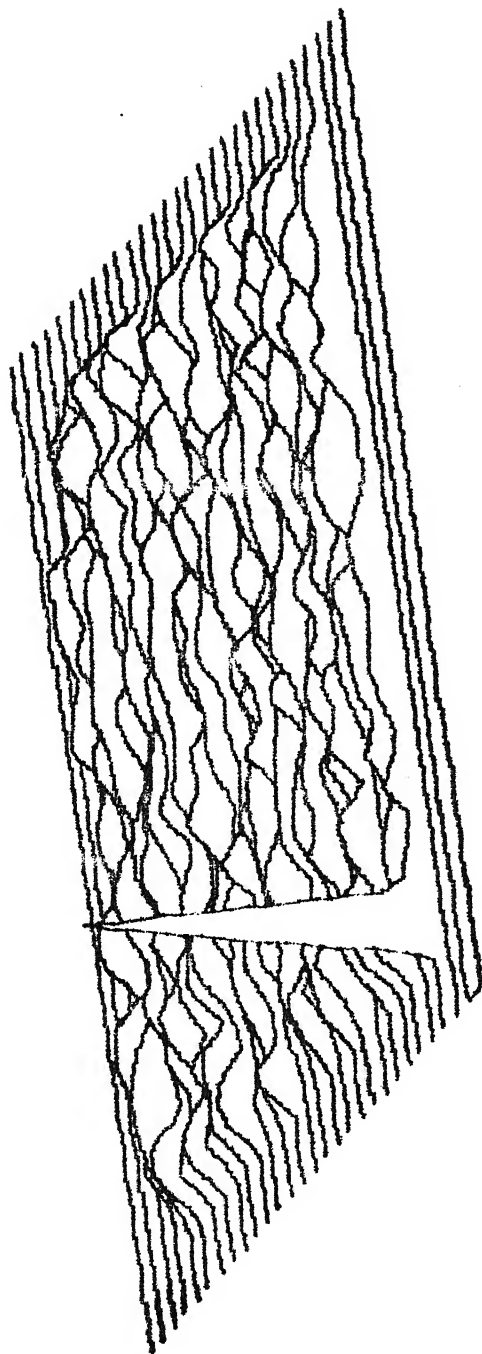


Fig. 2.5: The Autocorrelation Function of the Residual
for AR(1,1) Process.

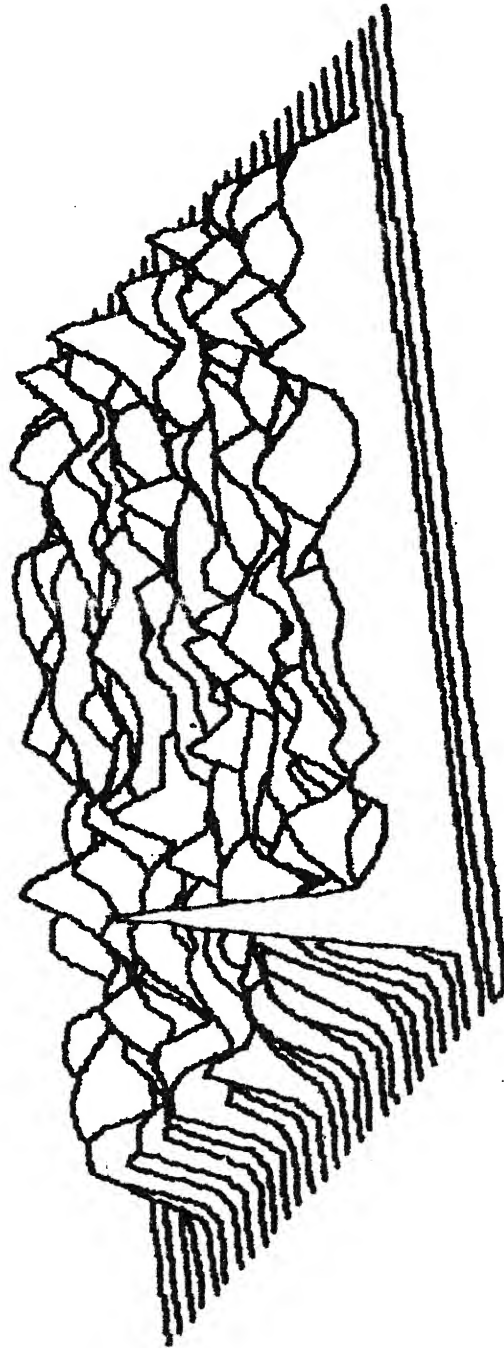


Fig. 2.6: The autocorrelation function of the residual for MA(1,1) process.

CHAPTER 3

FREQUENCY DOMAIN CHARACTERIZATION OF STOCHASTIC TEXTURES

3.1 INTRODUCTION:

In the last chapter, we have studied the spatial domain characterization of textures by standard two dimensional time series models. It was also stated that models based on either autoregressive or moving average processes were computationally efficient. In general, parameter estimation for ARMA process is a complicated procedure. In this chapter, we shall study another procedure for characterizing the texture. This procedure represents the system in terms of amplitude and phase response. This approach may be thought of as a frequency domain equivalent of time series model.

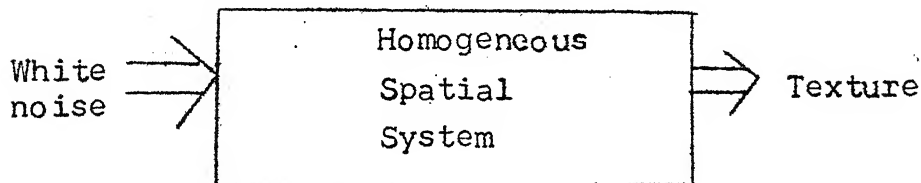


Fig. 3.1: A model for stochastic texture

The Figure 3.1 shows the model of texture generation. Like the two dimensional time series model, here also we assume that stochastic texture is generated by passing a white noise sequence through a linear homogeneous spatial

system. In our study, the 2D system is a 2D digital filter. In literature, the white noise sequence is generally taken as the computer generated random number sequence or IID sequence, having some specified probability distribution. But, in the frequency domain characterization of textures we use a 2D pseudo random array as the input noise array. Pseudo random sequences (generated from shift register sequences) have the property that they show nearly ideal autocorrelation function when defined in a cyclic fashion and they are also periodic. So, when such a doubly periodic array is passed through a linear system, the output also becomes periodic. So computation of amplitude and phase response can be done by FFT techniques. The useful concept for such system characterization is the moments of different orders. This method of texture modelling was tested here by generating a synthetic texture by such a process. For this simulation, nonbinary PN array was used as noise input.

3.2 PN SEQUENCES AND ARRAYS:

In this section, we will study the relevant properties of pseudo random sequences and its generation. It is found [31] that sequence, when generated as a maximal length sequence, shows some good properties. Such binary sequences, if they are generated by a m length shift register, will be periodic sequences of period $2^m - 1$. If we compute the

cyclic autocorrelation of such an n length sequence ($n=2^m-1$), the the autocorrelation follows the given relation

$$\begin{aligned} \rho(0) &= 1 \\ \rho(i) &= -\frac{1}{n}, \text{ for } 1 \leq i \leq n-1 \end{aligned} \quad (3.1)$$

where $\rho(\cdot)$ is the normalized autocorrelation function. Such a binary PN sequence approximates a white noise sequence. because, for a large length sequence, $\rho(i), \forall i \neq 0$ becomes negligibly small, which leads to a periodic delta function structure of autocorrelation. Such sequences have been used extensively in digital data scrambling, synchronisation etc., and a considerable body of literature exists [32,33,34].

Recently, several applications [] have called for two dimensional arrays whose 2D autocorrelation function is such that $\rho(0,0) = 1$ and $\rho(i,j)$ is small for $(i,j) \neq (0,0)$. It is very easy to generate such arrays from a given PN sequences [31]. For generating such an array of dimension $n_1 \times n_2$ with

$$\begin{aligned} \rho(0,0) &= 1 \\ \rho(i,j) &= -\frac{1}{n_1 n_2}, \quad \begin{aligned} &0 \leq i \leq n_1 - 1 \\ &0 \leq j \leq n_2 - 1 \\ &(i,j) \neq (0,0) \end{aligned} \end{aligned} \quad (3.2)$$

we need a sequence of length n , where $n = n_1 n_2 = 2^m - 1$, provided n_1 and n_2 are relatively prime. For generating a

31x33 array (as in our case), we need a PN sequence of length 1023. Since the generated sequence should be maximal length sequence in a specific Galois field [], we have to take the required primitive polynomial in that field. Primitive polynomial in a shift register sequence determines the governing difference equation. For the generation of a n -length shift register sequence ($n=2^m-1$), we need m initial values to start with. So, given the initial values and a primitive polynomial in the specified field, we can always generate a maximal length sequence. The most important property of such maximal length sequence (MLS) is its autocorrelation function. The autocorrelation function $f(i)$ of a real (or complex valued) sequence $s_0, s_1, s_2, \dots, s_{n-1}$ of length n is defined in cyclic sense as

$$f(i) = \frac{1}{n} \sum_{j=0}^{n-1} s_j \bar{s}_{j \oplus_n i} \quad (3.3)$$

$$i = 0, \pm 1, \pm 2, \dots$$

where bar represents complex conjugate and $j \oplus_n i$ means summation modulo- n . This autocorrelation function $f(i)$ is a periodic sequence, i.e., $f(n+i) = f(i)$. The autocorrelation function is shown in Fig. 3.2. The autocorrelation function of a pseudo-random sequence of length $n = 2^m - 1$ is given by

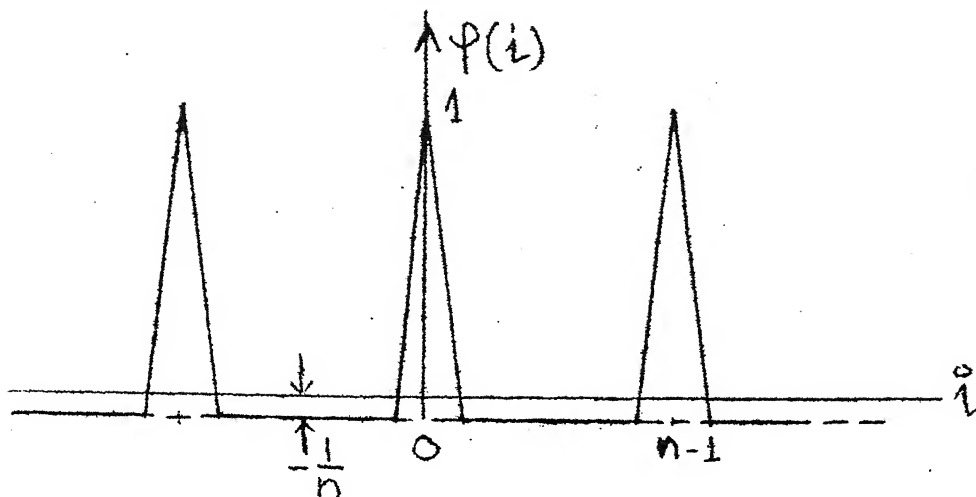


Fig. 3.2: Cyclic ACF function for a PN sequence of length n

$$\phi(0) = 1$$

$$\phi(i) = -\frac{1}{n} \quad 1 \leq i \leq 2^m - 2$$

Instead of binary PN sequence, having entries from 0 and 1 only, we can also have a nonbinary sequence, taken from an alphabet of q symbols, for any q which is a prime or power of a prime.

Let q be any prime power and let $GF(q)$ be the Galois field with q elements. We need a polynomial $g(x)$ of degree m with coefficients from $GF(q)$ which i) is not the product of two such polynomials of lower degree, and ii) has a zero as a primitive element of $GF(q^m)$, say ξ . We call such a polynomial primitive over $GF(q)$. In case of $GF(4)$ ($=GF(2^2)$) we have the elements $\{0, 1, \omega, \omega^2\}$ where ω is a root of the primitive polynomial of degree 2 in $GF(2)$. If we take such polynomial as $x^2 + x + 1$ then, since ω is root of it, $\omega^2 + \omega + 1 = 0$ and $\omega^3 = 1$.

Suppose, the primitive polynomial is

$$g(x) = x^m + h_{m-1}x^{m-1} + \dots + h_1x + h_0 \quad (3.3)$$

with $h_i \in GF(q)$, $h_0 \neq 0$. The shift register specified by the above polynomial is as follows. In the Figure 3.3, the

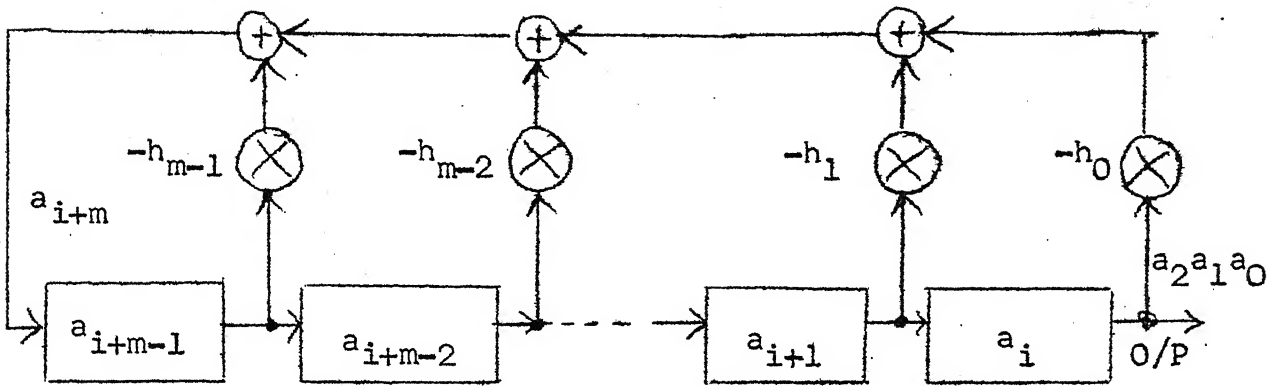


Fig. 3.3: General scheme for PN sequence generation

boxes contain elements of $GF(q)$, say a_{i+m-1}, \dots, a_i . The feedback path then forms as

$$a_{i+m} = -h_{m-1}a_{i+m-1} - h_{m-2}a_{i+m-2} - \dots - h_1a_{i+1} - h_0a_i \quad (3.4)$$

Eqn. (3.4) gives the recurrence relationship describing the output sequence. This is an infinite sequence of period $q^m - 1$. A segment of such sequence of length $q^m - 1$ is called the pseudo random sequence over $GF(q)$. As an example, if $q = 4$, $m = 2$ and $g(x) = x^2 + x + \omega$, we get the PN sequence

$$0 \ 1 \ 1 \ \omega^2 \ 1 \ 0 \ \omega \ \omega \ 1 \ \omega \ 0 \ \omega^2 \ \omega^2 \ \omega \ \omega^2 \quad (3.5)$$

of length $4^2 - 1 = 15$. This sequence can be modified to a sequence

$$0 \ 1 \ 1 \ 3 \ 1 \ 0 \ 2 \ 2 \ 1 \ 2 \ 0 \ 3 \ 3 \ 2 \ 3 \quad (3.6)$$

by replacing ω and ω^2 by 2 and 3 respectively.

To find the autocorrelation function of a nonbinary PN sequence, we have to get a complex valued sequence from the real one in order to use eqn. (3.3). The unique transformation (from real to complex) which gives the autocorrelation as in binary sequence is as follows.

If q is a prime or prime power then the complex sequence a is obtained from a by replacing each $\gamma \in GF(q)$ by $\exp [V-1 \frac{2\pi\gamma}{q}]$. After getting such complex sequence if we use the eqn. (3.3) then we get,

$$f(0) = 1, f(i) = \frac{-1}{q^m - 1} \quad 1 \leq i \leq q^m - 2$$

3.3 PSEUDO-RANDOM ARRAYS: GENERATION:

Though there are umpteen number of ways of making a matrix out of a vector, only one method out of them is useful in making pseudo random arrays so that the array has properties similar to that of the generating sequence. In the context of pseudo random arrays it is needed that

this 2D array will have same autocorrelation structure as that of pseudo random sequence.

First, we will consider the binary sequence and corresponding array. A binary (modulo-2) PN array is constructed by folding PN sequence. Let us take number of the form $n=2^{k_1 k_2}-1$ such that $n_1 = 2^{k_1}-1$ and $n_2 = n/n_1$.

Examples are

$$n=15=2^4-1 \text{ with } k_1=k_2=2 \quad n_1=3, n_2=5$$

$$n=63=2^6-1 \text{ with } k_1=3, k_2=2 \quad n_1=7, n_2=9$$

Cyclic autocorrelation of such array $(n_1 \times n_2)$ is given by

$$P(0,0) = 1$$

$$P(i,j) = -\frac{1}{n_1 n_2}, \forall i,j, (i,j) \neq (0,0)$$

The condition to be fulfilled is that n_1 and n_2 are relatively prime. Here we will explain the construction procedure of such arrays.

The starting point is a PN sequence $\underline{a} = (a_0, a_1, \dots, a_{n-1})$ obtained from a primitive polynomial of degree $m=K_1 K_2$. We use \underline{a} to fill up an $n_1 \times n_2$ array by writing \underline{a} down the main diagonal and continuing from the opposite side whenever an edge is reached. Figure 3.4 shows the construction of such array from a PN sequence when $n=15, n_1=3$ and $n_2=5$. Thus the PN sequence

0 0 0 1 0 0 1 1 0 1 0 1 1 1 1

(3.7)

produces the array

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |

(3.8)

Start at
north-west
corner and
move south-
east

| | | | | |
|-------|-------|-------|-------|--|
| a_0 | | | a_3 | |
| | a_1 | | | |
| | | a_2 | | |

Continue from
opposite side
when an edge is
reached

| | | | | |
|-------|-------|-------|-------|-------|
| a_0 | | | a_3 | |
| | a_1 | | | a_4 |
| a_5 | | a_2 | | |

The complete
array

| | | | | |
|----------|----------|----------|----------|----------|
| a_0 | a_6 | a_{12} | a_3 | a_9 |
| a_{10} | a_1 | a_7 | a_{13} | a_4 |
| a_5 | a_{11} | a_2 | a_8 | a_{14} |

Fig. 3.4: Construction of a PN array

These arrays are doubly periodic with period n_1 (horizontally) and n_2 (vertically). For such periodic array we can also define 2D cyclic autocorrelation function as

$$R(k, \ell) = \frac{1}{n_1 n_2} \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} x(i, j) \bar{x}(i \oplus_{n_1} k, j \oplus_{n_2} \ell) \quad (3.9)$$

$$\forall k = 0, 1, \dots, (n_1-1) \\ \text{and } \forall \ell = 0, 1, \dots, (n_2-1)$$

where $x(i, j)$ represents the PN array and \oplus_n is addition modulo n .

For such a PN array, it is found that

$$R(0, 0) = 1 \quad (3.10) \\ \text{and, } R(k, \ell) = -\frac{1}{n_1 n_2}, \forall (k, \ell) \neq (0, 0)$$

So, for large values of n_1 and n_2 , the autocorrelation will be an approximate two dimensional delta function which will give an approximate flat power spectral density. Same procedure holds for generating non-binary sequences.

3.4 SPATIAL FILTER MODEL

In this section, we will try to model a texture as a realization of a \angle ^{homogeneous} spatial process. Though it is



a standard procedure to create stochastic texture from computer generated random numbers, in this thesis we have studied textures generated from PN arrays. We make the assumption that the process is homogeneous and ergodic. Though homogeneity is not a valid assumption for real images, it may be a good one for texture type images. The spatial process or the texture is the output of a homogenous spatial filter. Most of the time series models assume that the white noise sequence is a Gaussian one, but here we do not make any such assumption. In fact, most of the real life textures are non-Gaussian. Since modelling of a random image means finding the system which will generate the image when excited by white noise, we can represent the system in two ways, i) in sample domain and ii) in frequency domain. This particular model represents a texture (or homogenous image) in terms of the frequency domain characterization of the system.

3.5 TEXTURE ANALYSIS:

In this section we will show that if a texture can be represented by the proposed model in Fig. 3.5, then it is

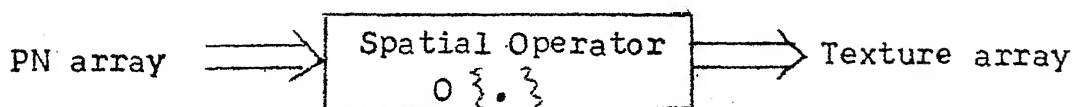


Fig. 3.5: Proposed Model

possible to compute all the filter parameters and then by inverse filtering we can get back the input noise parameters.

a) Set of Texture Parameters:

When a texture is generated or is supposed to be generated by the above model, then the resulting parameter sets for the textures are

- i) the texture mean
- ii) the histogram of the input white noise
- iii) the transfer function $H(z)$ of the filter (or its impulse response).

The second subset is equivalent to the set of different moments of the input IID sequence. These moments may be defined as B_1, B_2, \dots, B_n where

$$\begin{aligned} B_n &= \text{nth order moment} \\ &= E[b^n(\underline{k})] \end{aligned}$$

where $b(\underline{k})$ is the input noise.

This set of moments and histogram of the noise are equivalent in that sense any one of them will completely characterize the noise sequence.

3.6 PARAMETER EXTRACTION [18]

In this scheme of image modelling, the process of parameter extraction can be given in block form as in Fig.3.6.

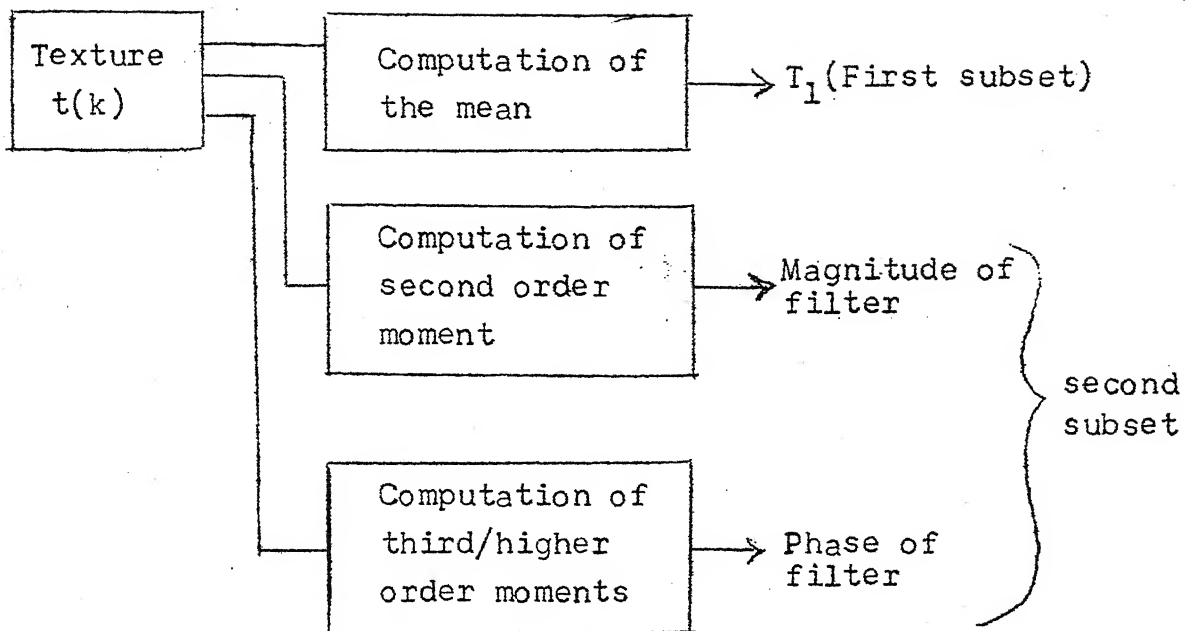


Fig. 3.6: Block diagram for parameter estimation

3.6.1 Amplitude Response Estimation:

Let us assume $t(k_1, k_2)$ be the pixel intensity at the point (k_1, k_2) . We also assume that input array $b(k_1, k_2)$ is doubly periodic with period N_1 and N_2 in horizontal and vertical directions respectively. Let us assume that $h(m, n)$ represents the impulse (2D) response of the filter. So given $h(m, n)$ and $b(k_1, k_2)$, we can write the output $t(k_1, k_2)$ as the 2D convolution of $h(m, n)$ and $b(k_1, k_2)$ as

$$t(k_1, k_2) = h(m, n) ** b(k_1, k_2)$$

where $**$ represents the 2D convolution

or,

$$t(k_1, k_2) = \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} b(m_1, m_2) h(k_1 - m_1, k_2 - m_2) \quad (3.11)$$

Here we can also define 2-dimensional Z-transform of an array $x(i, j)$ as

$$X(z_1, z_2) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} x(i, j) z_1^{-i} z_2^{-j} \quad (3.12)$$

So if we define $T(z_1, z_2)$, $H(z_1, z_2)$ and $B(z_1, z_2)$ as the 2D Z-transforms of the array $t(k_1, k_2)$, $h(k_1, k_2)$ and $b(k_1, k_2)$ then from eqn. (3.11) we get,

$$T(z_1, z_2) = H(z_1, z_2) \cdot B(z_1, z_2) \quad (3.13)$$

Though the first order moment or mean of texture $t(k_1, k_2)$ is defined statistically as

$$\bar{t}(k_1, k_2) = E[t(k_1, k_2)]$$

it is estimated as

$$\bar{t}_e(k_1, k_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} t(k_1, k_2) \quad (3.14)$$

The second order moment of the texture $t(k_1, k_2)$ gives the autocorrelation function array $M_2(\ell_1, \ell_2)$ of the texture. The second order moment of $t(k_1, k_2)$ is defined again in statistical manner as

$$M_2(l_1, l_2) = E[t(k_1, k_2)t(k_1+l_1, k_2+l_2)] \quad (3.15)$$

Using eqn. (3.11) we get

$$M_2(l_1, l_2) = E\left[\sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} b^2(m_1, m_2) h(k_1-m_1, k_2-m_2) h(k_1+l_1-m_1, k_2+l_2-m_2)\right] \quad (3.16)$$

$$= B_2 \sum_{m_1=0}^{N_1-1} \sum_{m_2=0}^{N_2-1} h(m_1, m_2) h(m_1+l_1, m_2+l_2) \quad (3.17)$$

Eqn. (3.17) is true only if input array is a white one.

Here B_2 represents the 2nd order moment of the input noise. For time being, we can assume $B_2 = 1$.

If we take the 2D Z-transform of $M_2(l_1, l_2)$, we get

$$M_2(z_1, z_2) = \sum_{l_1=-\infty}^{\infty} \sum_{l_2=-\infty}^{\infty} M_2(l_1, l_2) z_1^{-l_1} z_2^{-l_2} \quad (3.18)$$

$$M_2(z_1, z_2) = H(z_1, z_2) \cdot H(z_1^{-1}, z_2^{-1}) \quad (\text{by using eqn. (3.17)}) \quad (3.19)$$

$$M_2(z_1, z_2) = |H(z_1, z_2)|^2 \quad (3.20)$$

So we see, in 2D case, the 2D Z-transform of the second order moment of the output of a system gives the amplitude square response of the system.

If the process is Gaussian then the 2nd order moment and first order moment contain all the statistical information of the process. By inverse filtering with a zero-phase filter we can get back the white noise. In that case, it is not necessary to compute the higher order moments. But in case the process is not Gaussian, it is possible to estimate the phase of the filter $H(z_1, z_2)$ from higher order moments.

3.6.2 Phase Computation:

Phase information is contained in the non-Gaussian part of the higher order moments. In the simplest non-Gaussian case we can assume that the process is not Gaussian at the third order i.e. $B_3 \neq 0$. We can define third order moment as

$$M_3(k_1, k_2; l_1, l_2) = E[t(0,0)t(k_1, k_2)t(l_1, l_2)] \quad (3.21)$$

It shows that the general filter structure gives a four dimensional third order moment function. In order to avoid long computation we describe the process of phase estimation by considering a special filter - called separable filter. If a 2D filter is a separable one, then it can be thought of as a combination of two one-dimensional filters. In case of separable filter, the 2D impulse response $h(m, n)$ can be written as

$$h(m, n) = h_1(m)h_2(n) \quad (3.22)$$

or in frequency domain,

$$H(z_1, z_2) = H_1(z_1)H_2(z_2) \quad (3.23)$$

where $h_1(m) \Leftrightarrow H_1(z_1)$

and $h_2(n) \Leftrightarrow H_2(z_2)$

We will show here how to use the concept of third order moment in the estimation of phase of a unidimensional filter. For this purpose let us consider the Fig. 3.7.

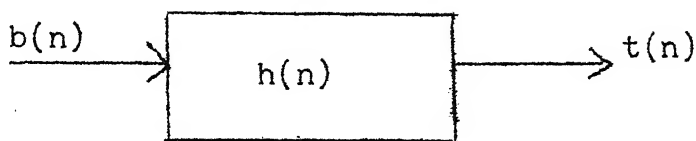


Fig. 3.7: One dimensional model

Let us define the third order moment of $t(n)$ to be $M_3(l_1, l_2)$ where

$$M_3(l_1, l_2) = E[t(0)t(l_1)t(l_2)] \quad (3.24)$$

$$M_3(l_1, l_2) = B_3 \sum_{m=0}^{N-1} h(m)h(m+l_1)h(m+l_2) \quad (3.25)$$

Here we assume $b(n)$ and hence $t(n)$, to be $\angle N$ -periodic. Taking Z-transform of $M_3(l_1, l_2)$ we get

$$M_3(z_1, z_2) = \sum_{l_1} \sum_{l_2} M_3(l_1, l_2) z_1^{-l_1} z_2^{-l_2} \quad (3.26)$$

Using eqn. (3.25), we get

$$\hat{M}_3(z_1, z_2) = B_3 H(z_1) H(z_2) H[(z_1, z_2)^{-1}] \quad (3.27)$$

Let us assume $z_1 = e^{j\omega_1}$ and $z_2 = e^{j\omega_2}$, then from eqn. (3.27) we get,

$$\phi_3(\omega_1, \omega_2) = \epsilon \pi + \phi(\omega_2) - \phi(\omega_1 + \omega_2) \pmod{2\pi} \quad (3.28)$$

where

$$\phi_3(\omega_1, \omega_2) = \text{phase of } \hat{M}_3(z_1, z_2)$$

$$\phi(\omega) = \text{phase of } H(z_1)$$

$$\epsilon = 0 \text{ or } 1 \text{ depending on sign of } B_3.$$

So, after computing the phase of $\hat{M}_3(z_1, z_2)$, we have to find the phase of the filter $H(z)$ i.e. $\phi(\omega)$ by using eqn. (3.28). Here we will give an iterative procedure to compute the phase.

Let us assume that we want phase $\phi(\omega)$ of an array of N values, depending on the size of the filter.

$$\text{Let us define } \omega_1 = \frac{2\pi(k-1)}{N}$$

$$\text{and } \omega_2 = \frac{2\pi(l-1)}{N}$$

Hence from eqn. (3.28) we get

$$\phi_3(k, l) = \pi + \phi(k) + \phi(l) - \phi(k+l-1) \pmod{2\pi} \quad (3.29)$$

The eqn. (3.29) is the digitized version of eqn. (3.28).

For $\ell = 2$ in eqn. (3.29), we get,

$$\phi_3(k, 2) = \pi + \phi(k) + \phi(2) - \phi(k+1) \pmod{2\pi} \quad (3.30)$$

$$\phi_3(1, 2) = \pi + \phi(1) + \phi(2) - \phi(2) \quad (3.31)$$

$$\phi(1) = \phi_3(1, 2) \quad (3.31)$$

assuming $\epsilon = 1$

Similarly,

$$\phi(2) = \frac{1}{N} \sum_{k=1}^N \phi_3(k, 2) \quad (3.33)$$

We can rewrite eqn. (3.30) (with $\epsilon = 1$) as

$$\phi(k+1) = \phi(k) + \phi(2) - \phi_3(k, 2) \quad (3.34)$$

We will use eqn. (3.29) to eqn. (3.34) for the phase estimation of the 2D filter.

3.7 USE OF SEPARABLE FILTER TO COMPUTE PHASE

We have already stated that if the linear filter is not a separable one then third order moment yields a four-dimensional function and from such function phase computation is a very time consuming process. For this reason, we have taken a separable filter. In such cases, the convolution also becomes very convenient one.

A filter $h(i,j)$ is said to be a separable one if we can write

$$h(i,j) = h_c(i)h_r(j) \quad (3.35)$$

$$\text{in Z-domain } H(z_1, z_2) = H_C(z_1)H_R(z_2) \quad (3.36)$$

If we say $X(z_1, z_2)$ is the Z-transform of the input and $Y(z_1, z_2)$ is the Z-transform of the output, then entire process can be shown as in Fig. 3.8.

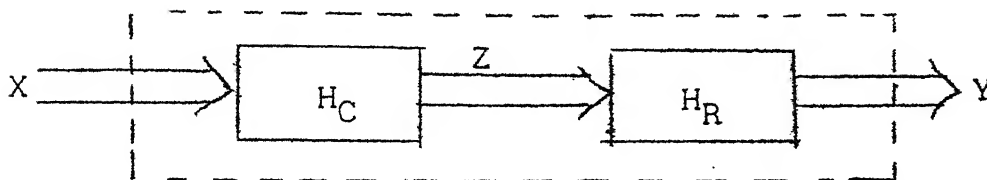


Fig. 3.8: A Separable filter structure

Though we can get \underline{Y} by 2D convolution of \underline{X} with \underline{H} , still we can utilise the separable nature of the filter. In the Fig. 3.8 the filters H_C and H_R are both 1D filter. Let us assume \underline{X} is a $n_1 \times n_2$ array. Then the first filter H_C will take each column of \underline{X} in succession and will modify them to get another intermediate array \underline{Z} . Each column of \underline{Z} is the column filtered version of \underline{X} . This new array \underline{Z} is fed to the next 1D filter H_R . H_R will take each row of \underline{Z} and it will modify the row to produce a row of the array \underline{Y} . So \underline{Y} is the row-filtered version of the array \underline{Z} .

This separable property of the system can be exploited in computing the phase. If $H_R(\theta_1)$ represents the phase of the 1D filter H_R and $H_C(\theta_2)$ represents the 1D phase of H_C then phase of the total filter H is given by

$$H(\theta_1, \theta_2) = H_R(\theta_1) + H_C(\theta_2) \pmod{2\pi} \quad (3.37)$$

So effectively the 2D phase of the system is computed by the help of its one-dimensional portions.

3.8 SIMULATION RESULTS:

In the simulation process we have taken a 31×33 doubly periodic pseudo random array and a 2D filter with transfer function $H(z_1, z_2)$ given by

$$H(z_1, z_2) = \left(\frac{2-z_1}{1-2z_1} \right) \left(\frac{2-z_2}{1-2z_2} \right) \quad (3.38)$$

This filter is a constant magnitude (all pass) filter. It has the property of introducing no correlations. By replacing $z_1 = e^{-j\omega_1}$ and $z_2 = e^{-j\omega_2}$ we can compute the amplitude response of such filter on the unit circle. Then

$$H(\omega_1, \omega_2) = \frac{(2-e^{-j\omega_1})}{(1-2e^{-j\omega_1})} \cdot \frac{(2-e^{-j\omega_2})}{(1-2e^{-j\omega_2})} \quad (3.39)$$

$$\begin{aligned} \text{Since, } e^{-j\omega_1} &= \cos \omega_1 - j \sin \omega_1 \\ \text{and } e^{-j\omega_2} &= \cos \omega_2 - j \sin \omega_2 \end{aligned} \quad (3.40)$$

simple trigonometric identities lead to

$$|H(\omega_1, \omega_2)| = 1. \quad (3.41)$$

This type of filter shows a pole-zero structure in its transfer function which in sample domain, is equivalent to an ARMA process. Since $H(z) = \frac{N(z)}{D(z)}$ where $N(z)$ and $D(z)$ are polynomials in z such filter is an IIR (infinite impulse response) filter. But impulse response of such filter is also a decaying one in both direction. The input white noise array was taken to be 31×33 from $GF(4)$, i.e. the elements of the array are from the set $\{0, 1, 2, 3\}$. Since total number of elements in this array is $1023 (= 4^5 - 1)$ we need a shift register of length 5. We also need a primitive polynomial of degree 5 in $GF(4)$. The polynomial is taken to be $g(x)$

$$g(x) = x^5 + x + \omega \quad (3.42)$$

where ω is a root of the primitive polynomial of degree 2 in $GF(2)$. If we take that polynomial as $x^2 + x + 1 = 0$ then

$$\begin{aligned} \omega^2 + \omega + 1 &= 0 \\ \text{and } \omega^3 &= 1 \end{aligned} \quad (3.43)$$

So the required shift register is as shown in Fig. 3.9.

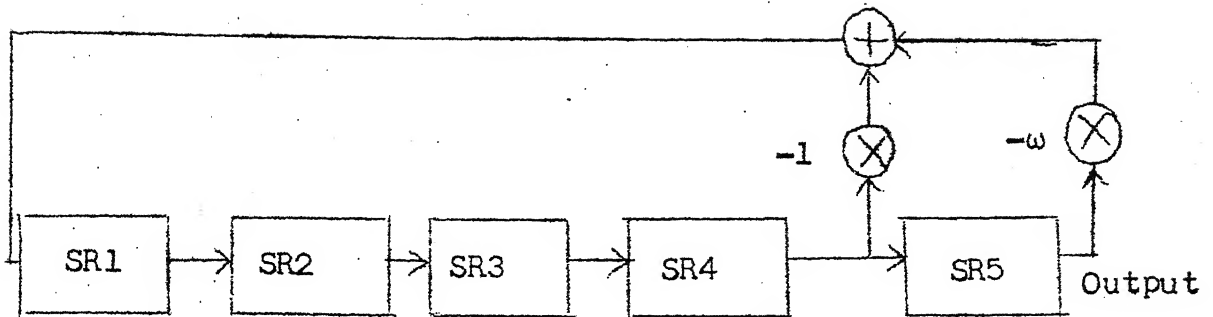


Fig. 3.9: The feedback connection for Eqn. (3.42).

After generating the periodic sequence of length 1023 we then used the scheme for converting a vector into a matrix to form a 31×33 array. This 2D array x (say) was then filtered by the spatial system $H(z_1, z_2)$ to get the periodic output which was the required texture. Though we have represented the filter in Z-domain by the transfer function $H(z_1, z_2)$, the filtering scheme was developed in sampling domain by writing the corresponding difference equation. We also took advantage from the separable property of the filter. Here we refer to the Fig. 3.8. For the given system

$$\begin{aligned}
 H(z_1, z_2) &= \left(\frac{2-z_1}{1-2z_1} \right) \cdot \left(\frac{2-z_2}{1-2z_2} \right) \\
 &= H_C(z_1) H_R(z_2)
 \end{aligned}$$

If we assume,

$$H_C(Z_1) = \frac{Y'(Z_1)}{X'(Z_1)} = \frac{2-Z_1}{1-2Z_1} \quad (3.44)$$

then, by taking Z-transform of eqn. (3.44), we get,

$$y'(n) = 0.5 x'(n) - x'(n-1) + 0.5 y'(n-1) \quad (3.45)$$

As mentioned earlier, H_C will take each column of the input and produces a new column following eqn. (3.45). If z represents the 2D output of H_C then,

$$z(i,j) = 0.5x(i,j) - x(i,j-1) + 0.5z(i,j-1) \quad (3.46)$$

Similar expression is obtained in the case of second filter H_R ;

$$y(i,j) = 0.5z(i,j) - z(i-1,j) + 0.5y(i-1,j) \quad (3.47)$$

In using equation (3.46) and eqn. (3.47) it is to be remembered that the input is periodic and system is causal, i.e.,

$$x(i,j) = 0, \quad \forall i \leq 0, j \leq 0$$

and

$$y(i,j) = 0, \quad \forall i \leq 0, j \leq 0$$

For computing the impulse response array, we excited the system by a 2D delta function $\delta(i,j)$. For the computation of 2D Z-transform (on the unit circle) of second

and third order moments of the output, we have used fast fourier transform routine FFTP (IMSL). Fig. 3.10 through 3.13 depict the 2D cyclic autocorrelation of the input, the actual and estimated amplitude and phase response respectively of the all pass system used in the study. In such cases we cannot take input noise array of arbitrary dimension. In order to maintain the idealness of autocorrelation, we have to arrange the elements of the sequence in a particular way to get the array. Considering all these facts, it seems that for such simulation purposes, computer generated random number are more convenient than shift register sequences.

Such pseudo random sequences/arrays, can however still be used as a substitute for computer generated random number for many applications. The periodic structure and almost ideal autocorrelation of pseudo random numbers help us in this respect.

Since input noise is a pseudo-random array, the generated texture should be named 'pseudo random' texture.

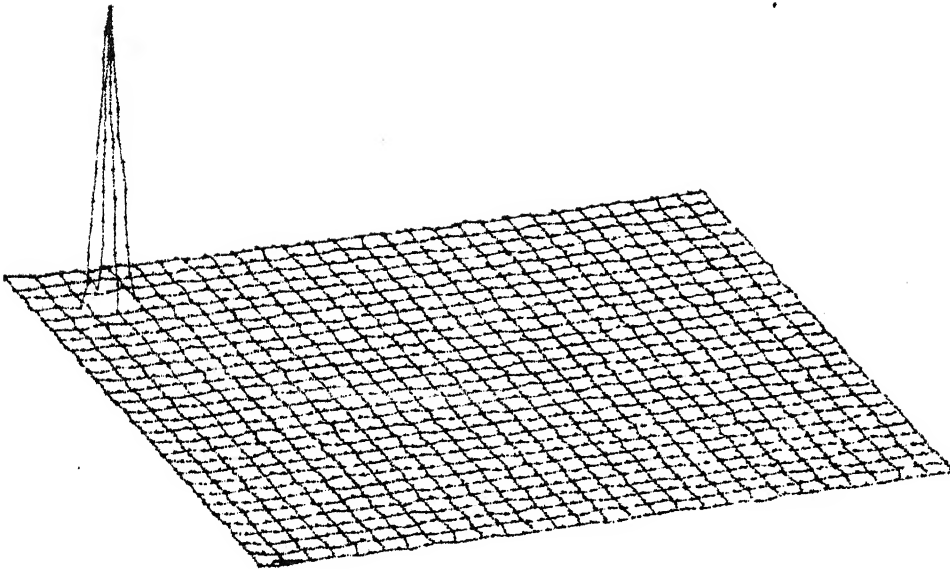


Fig. 3.10: The 2D cyclic autocorrelation of PN-array

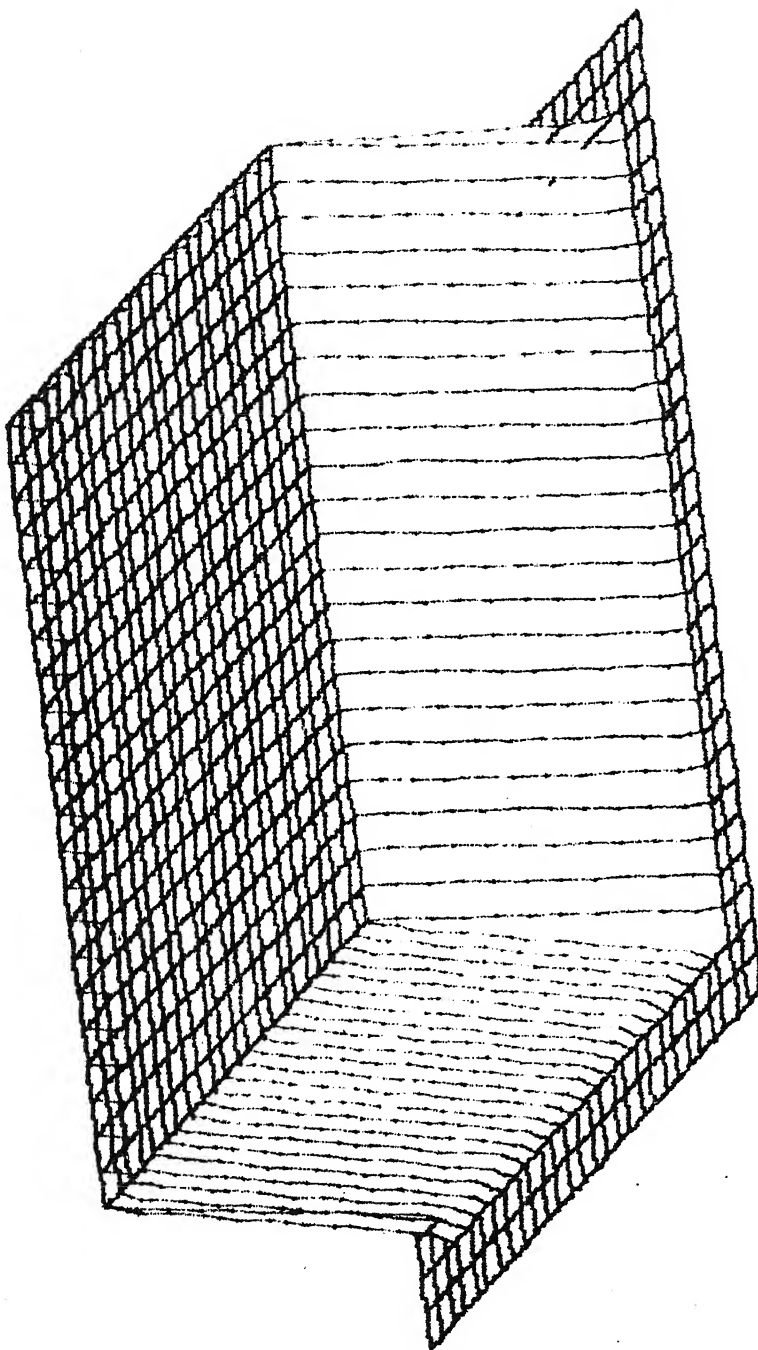


FIG. 3.11: The Actual Amplitude response of the filter used in Simulation.

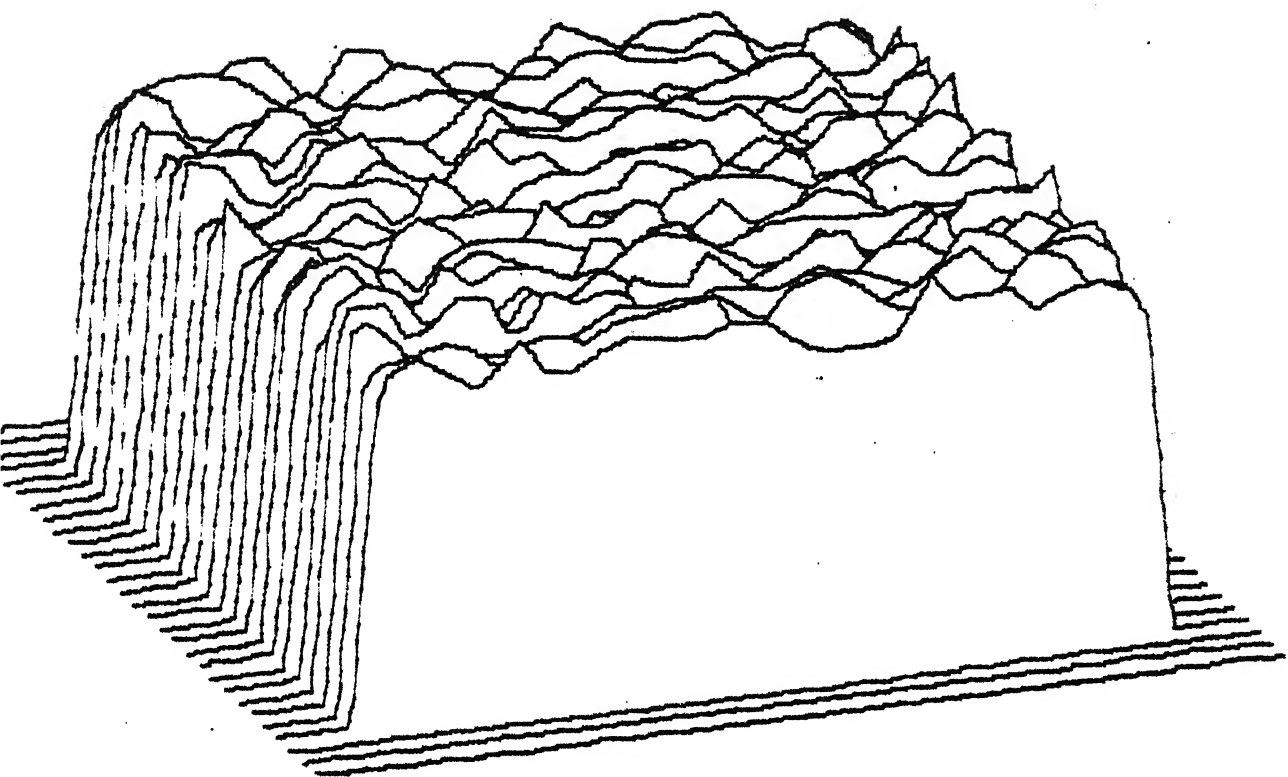


Fig. 3.12: The estimated amplitude response of the filter used in simulation.

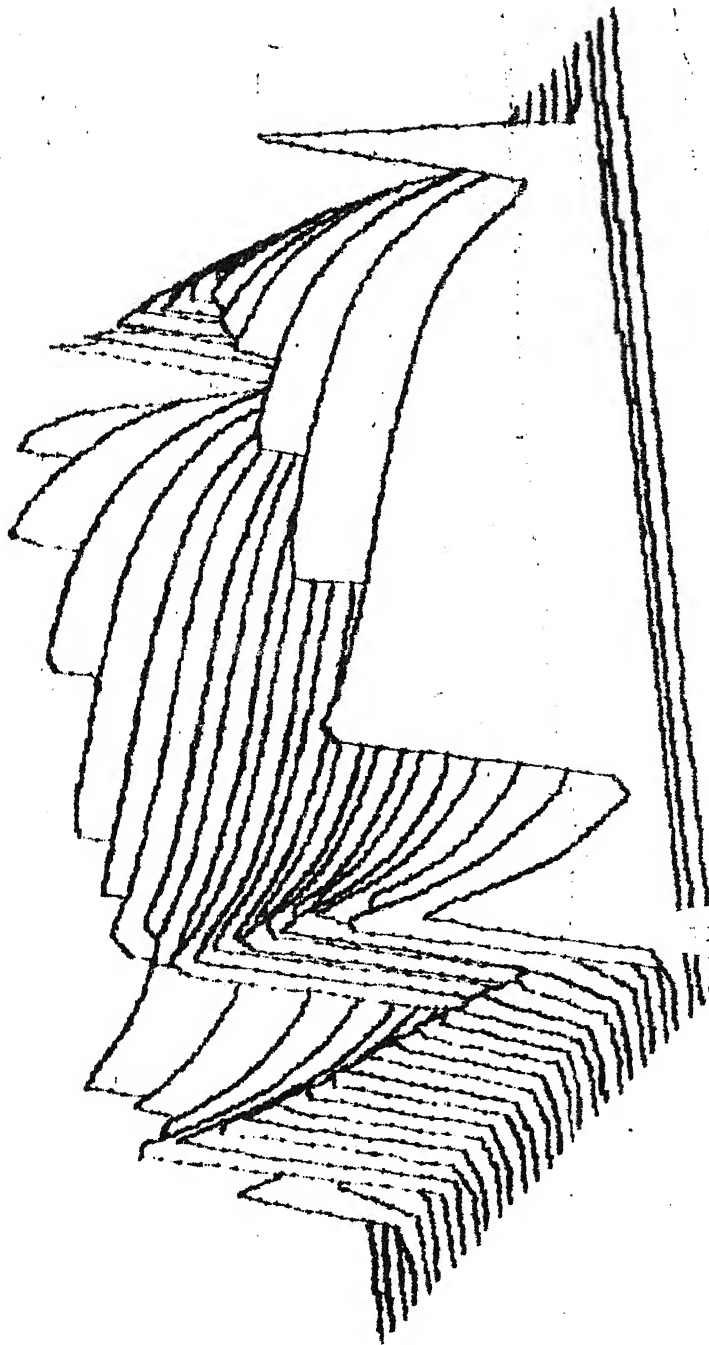


Fig. 3.13: The actual phase response of the filter used in simulation.

CHAPTER 4

ANALYSIS AND SYNTHESIS OF TEXTURES BY NONCAUSAL RF MODELS

4.1 INTRODUCTION

In the last two chapters, we have mentioned sample domain and frequency domain characterization of textures. In this chapter, we will study a new class of RF models where the concept of neighbourhood is a general one, i.e., it can also take of non-causal neighbourhood. These models may be called as non-causal spatial interaction model as described by Kashyap [10]. Such RF models represent an image as a distribution of gray levels over a finite or infinite lattice where each point is supposed to be equidistant.

Every such model is characterized by a set of neighbours, the corresponding coefficients and an independent identically distributed random sequence with prespecified probability density. Typically an image is characterized by 2D scalar data, the gray level variations defined over a rectangular or square lattice. In such lattice structures, gray level at a particular lattice point has a statistical dependency on the points inside a specified neighbourhood.

There are two nonequivalent ways of specifying the interaction between pixels in a given lattice, leading to

two different classes of models, the simultaneous models [22] and conditional Markov models. Here we will discuss only the spatial interaction models over square lattice. These models do not require that the underlying field is Gaussian. These models are therefore quite general, i.e., a variety of image types can be generated by choosing different neighbour sets, IID sequences of different probability density functions and different parameter sets.

4.2 NOTATIONS

$r = (r_1, r_2)$, $s = (s_1, s_2)$: Co-ordinates of grid points, r_i and s_i integers. $y(s)$ = gray level or intensity at pixel position s , $y(s)$ is a non-negative, real number.

$$\Omega = \left\{ s = (s_1, s_2), 1 \leq s_1, s_2 \leq M \right\}$$

= set of grid points of a finite image of size $M \times M$.

$$I^2 = s = (s_1, s_2), -\infty \leq s_1, s_2 \leq \infty$$

$$y = \text{col} [y(1,1), y(1,2), \dots, y(M,M)]$$

= vector of a finite $M \times M$ image

$$N = \text{a neighbour set, a subset of } I^2 \text{ } (0,0) \notin N$$

$$z = (z_1, z_2), \text{ a pair of operators, defined below}$$

$$\text{If } r, s \in I^2, z^r y(s) \triangleq y(s_1 + r_1, s_2 + r_2)$$

$$\lambda^T = (\lambda_1, \lambda_2) = \text{a pair of frequencies in both directions.}$$

λ_i = frequency in i th direction

$$\exp(\sqrt{-1}\lambda) = (\exp\sqrt{-1}\lambda_1, \exp\sqrt{-1}\lambda_2)$$

$$A(z) = 1 - \sum_{r \in N} \theta_r z^r$$

$$A_D(r) = A(z = e^{\sqrt{-1}2\pi r/M})$$

$$Y(\cdot) = \text{D.F.T. of } \{y(s), s \in \Omega\}$$

$$U(\cdot) = \text{D.F.T. of } \{u(s), s \in \Omega\}$$

$R(\cdot)$ = Autocovariance function

$S(\lambda)$ = Spectral density

$p(\cdot)$ = Probability density

The convention for labelling members of N are as follows:

$x(-1,0)$ North

$x(0,-1)$ West

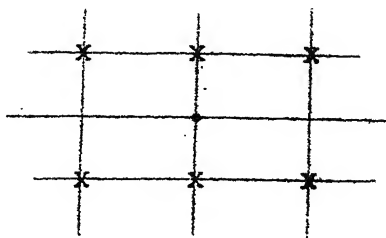
$x(0,1)$ East

$x(1,0)$ South

For example if the set N is represented by the coordinate as given

$$N = (-1,-1), (-1,0), (1,0), (1,-1), (1,0), (1,1)$$

then pictorially,



4.3 SUMMARY OF DISCRETE SPATIAL INTERACTION MODELS

In this chapter we will discuss only discrete spatial interaction models with gray levels either real values or integers. In all such models the input is a set of independent random variables $u(s)$ with a common density p_u . Such sequences are called independent identically distributed sequence (IID). The random field (RF) image $y(\cdot)$ is generated by transforming the input sequence in a suitable manner. In general, the system representation of such transformation may be of the following type .

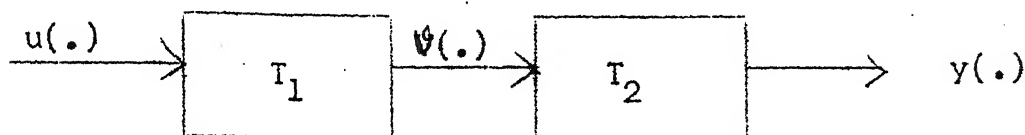


Fig. 4.1: A generalized model for RF Textures

Firstly, we will discuss about T_2 and then with proper context we will discuss T_1 . The system T_2 is characterized by transfer function $A(z)$ which is defined by a neighbour set N and postulated to be

$$A(z) = 1 - \sum_{r \in N} \theta_r z^r, \quad \theta_r \text{ real} \quad (4.1)$$

The system T_1 and T_2 may be interpreted as a difference equation in sample domain or multiplication in transform

domain. The output of the first system T_1 may be correlated or uncorrelated depending on the system T_1 . Now if we want to fit such a model for image data, then our main objective is to estimate the finite dimensional parameters $\{\theta_r\}$ and the probability $p_u(.)$ of the input IID sequence. We can divide the generalized model (Fig. 4.1) in two ways:

Case 1: The infinite lattice random field (ILRF) models, where $y(s)$ is defined over an infinitely extended grid or lattice, i.e., for such image $y(i,j)$ $-\infty < (i,j) < \infty$.

Case 2: The finite lattice random field (FLRF) model is that one where $y(s)$ is defined only over a finite region and in that case $\Omega = \{(i,j), 1 \leq i,j \leq M\}$ representing $M \times M$ image.

In general for RF models whether it is a infinite lattice or a finite lattice we can represent the image data y as

$$y(s) - \alpha = - \sum_{s_k \in N} \theta_k [y(s + s_k) - \alpha] + \sqrt{\sigma} u(s) \quad (4.2)$$

$$s = (i,j)$$

$$i,j = 0, \pm 1, \pm 2, \dots$$

Depending on the statistical nature of $u(s)$ we may obtain various types of RF models such as SAR (Simultaneous Autoregressive), CM (Conditional Markov) etc. One advantage of such RF models is that they incorporate geometrical and

statistical dependency of the pixel on its specified neighbourhood, not considered in many other models. By such models we can generate synthetic textures obeying a given model with known parameters by using a set of random numbers.

4.4 INFINITE LATTICE MODEL

Let us consider Fig. 4.1 as the general model. In IL models the transformation T_2 is interpreted as a difference equation whose coefficients are $\{\theta_r\}$ from $A(z)$ and

$$y(s) - \sum \theta_r y(s+r) = \sqrt{p} v(s) \quad (4.3)$$

This equation may also be written as

$$A(z) y(s) = \sqrt{p} v(s), \quad z = (z_1, z_2) \quad (4.4)$$

where z_i are regarded as delay operators in the 2 directions and $A(z)$ is given in equation (4.1). The different choices for T_1 lead to different types of models.

Simultaneous AR (SAR) Models:

For a SAR model defined over infinite lattice, we have

$$T_1 = 1 \text{ and } v(s) = u(s)$$

Simultaneous ARMA (SARMA) Models:

For a SARMA model defined over infinite lattice, we have,

$$T_1 = B(z)$$

$$= 1 + \sum_{r \in N} \phi_r z^r, \quad \{\phi_r\} \text{ are real}$$

Equivalently,

$$\begin{aligned} y(s) - \sum_{r \in N} \phi_r y(s+r) &= \sqrt{\phi} v(s) \\ &= \sqrt{\phi} [u(s) + \sum_{r \in N} \phi_r u(s+r)] \end{aligned}$$

Separable Models:

Here $A(z)$ and $B(z)$ possess a factorization

$$A(z) = A_1(z_1)A_2(z_2)$$

$$B(z) = B_1(z_1)B_2(z_2)$$

The neighbour sets characterizing $A(z)$ and $B(z)$ are located in a quadrant.

Conditional Markov Model:

In conditional Markov (CM) models, the neighbour set N , used in characterizing the polynomial A and B is symmetric, i.e., $r \in N \Rightarrow -r \in N$. Secondly the system T_1 is implicitly defined by the following relation.

$$\begin{aligned} S_v(\lambda) &= \text{Spectral density of } v(.) \\ &= |2[\exp\{-1/2\pi\lambda_1\}, \exp\{-1/2\pi\lambda_2\}]|^2 \end{aligned} \quad (4.5)$$

Equivalently, the sequence $v(\cdot)$ has the following covariance function.

$$\begin{aligned} R_v(s) &= 1 \quad \text{if } s = (0,0) \\ &= -\theta_s \quad \text{if } s \in N \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{4.6}$$

For such models, the coefficients $\{\theta_s\}$ follows the relation

$$\theta_s = -\theta_s$$

We can assume the relationship between $u(\cdot)$ and $v(\cdot)$ as follows

$$T_1 = \sqrt{B(z)} \tag{4.7}$$

To validate the assumption we can consider two mutually exclusive cases.

Case i: There exists a polynomial $A_1(z)$ so that

$$A(z) = A_1(z_1) A_1(z^{-1})$$

and

$$A_1(z) = 1 - \sum_{r \in N_1} \theta_r^1 z^r \tag{4.8}$$

In such a case T_1 in eqn. (4.7) is interpreted as a difference operator

$$v(s) = A_1(z) u(s) \tag{4.9}$$

So in that case, the conditional model reduces to a simultaneous model.

case ii: Let us assume that $A(z)$ has no factorization, so in that case eqn. (4.7) has no meaning.

It is also important to note that a symmetric polynomial $A(z_1)$ in a single variable z_1 is always factorable. The general nonfactorability of a symmetric 2D polynomial $A(z_1, z_2)$ is one of the key differences between the 1D and 2D theory.

4.5 FINITE LATTICE MODELS

These models were introduced by Kashyap [32] and they have many elegant properties. In FL models both the inputs and output (image) are assumed to be defined over a $M \times M$ grid.

As we are interested in finite lattice image so in the subsequent sections we will discuss only FL model, however, an equivalence of these with IL models will also be mentioned.

Neighbour Sets: Markov Definition and Stationarity

Neighbour Sets: A neighbour set N is any finite subset of the total set $s_1, s_2, -\infty \leq s_1, s_2 \leq \infty$. For all such sets $(0,0) \notin N$. A neighbour set is called symmetric if

$$(i,j) \in N \Rightarrow (-i,-j) \in N$$

The first order and second order neighbour sets are defined below.

```

      x
      .
x   .   x
      .
      x
  
```

N: First order

```

      x   x   x
      .
      x   .   x
      .
      x   x   x
  
```

N: Second order

In the last two chapters we have analysed texture where neighbourhood was assumed to be causal. A causal neighbourhood has all its members in a quadrant

$N = (i,j), (0,0) \quad N, i < 0, j < 0$

```

      x
      .
x   .   x
      .
      x
  
```

N: Causal

```

      x
      .
      x   .
      .
      x
  
```

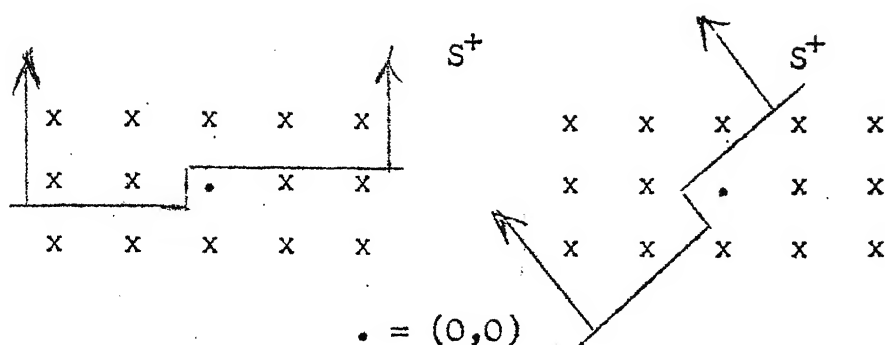
N: Semi causal

N is called as one-sided or unilateral if N is a finite subset of a half plane S^+ defined as

Definition: (Half Plane S^+): S^+ is recursively defined as follows

- i) $s_1 \in S^+, s_2 \in S^+ \Rightarrow s_1 + s_2 \in S^+$
- ii) $s \in S^+ \Rightarrow -s \notin S^+$
- iii) $(0,0) \notin S^+$

From the above definitions it is seen that S^+ is not unique. Some examples of S^+ are



The non-causal neighbourhood is a more general neighbourhood and their use is more realistic for discrete image data modelling.

Markov Definitions:

1D Markov Sequences:

A unidimensional sequence $\{y(t), t = 1, 2, \dots\}$ is said to be markov of order m if

$$p(y(t)|y(t_1), t_1 < t) = p(y(t)|y(t-1), \dots, y(t-m)) \quad (4.10)$$

where p is the ^{probability} density function. Here m is called the memory of the process. A 1D process $y(t)$ is called bilateral markov of order m if

$$\begin{aligned} p(y(t)|\text{all } y(t_1), t_1 \neq t) \\ = p(y(t)|y(t_{\pm 1}), y(t_{\pm 2}), \dots, y(t_{\pm m})) \end{aligned}$$

(4.11)

In a similar manner, we can define vector markov sequence. In such a case, every component of the vector is said to be projection of a markov process.

2D Markov Definitions [35]

A 2D sequence $y(s)$ is said to be bilateral markov with respect to a symmetric neighbourhood N if

$$p(y(s)|\text{all } y(s'), s \neq s') = p(y(s)|y(s+s'), \text{all } s' \in N) \quad (4.12)$$

Comparing eqn. (4.11) and eqn. (4.12) we see that bilateral markov in 2D is just a generalization of bilateral markov in 1D and in general bilateral markov in 2D is known as 2D markov sequence.

Stationarity and Isotropy:

A sequence $y(s)$ is covariance stationary if

Covariance $[y(s), y(s+r)] = \text{function of } r \text{ only}$

$$\triangleq R(r) = R(-r)$$

where covariance $[xy] \triangleq E[(X-E(x))(Y-E(y))]$

A sequence $y(.)$ is weak stationary if it is covariance stationary and

$$E[y(s)] = \mu, \forall s$$

The sequence $y(.)$ is called a strong stationary sequence if its probability density function obeys the following conditions.

$$p(y(s') = x_1, y(s'') = x_2) = f(x_1, x_2, s' - s'')$$

In this study by stationarity we will mean the weak stationarity in the covariance sense. A stationary sequence will be called isotropic if,

$$R(s = (i, j)) = R(|i|, |j|) = R(j, i)$$

For an infinite lattice covariance stationary R.F. the spectral density $S(\underline{\lambda})$ is defined as

$$\begin{aligned} S(\underline{\lambda}) &= \sum_{s \in \mathbb{I}^2} R(s) \exp(j 2\pi \underline{s} \cdot \underline{\lambda}) \\ &= \sum_{s_1 = -\infty}^{\infty} \sum_{s_2 = -\infty}^{\infty} R(s_1, s_2) \exp[j 2\pi (s_1 \lambda_1 + s_2 \lambda_2)] \end{aligned}$$

4.6 ANALYSIS OF FINITE LATTICE MODELS

We have discussed in the previous sections some of the properties of infinite lattice RF(ILRF) models. The greatest impediment to the use of RF models defined over infinite lattices for analysing finite images is the extensive computation involved. Even the computation of some of the basic quantities like $M^2 \times M^2$ covariance matrix for an $M \times M$ image requires extensive computations. Consequently, most of the RF models, used in image processing use simplistic assumptions like isotropy, causal neighbourhood etc., so as to reduce the computational complexity.

Of course, there is considerable interest in the approximation methods. However, every approximation method seems to be tailored to a particular RF model and a particular application.

The approximation of a covariance matrix by a circulant matrix or by block-circulant matrix is justified only by asymptotic considerations, i.e., the size of the image tends to infinity. These approximations are not suitable for synthesis of any image, obeying a given RF model.

Here in this section, we will analyse the basic features of finite or lattice RF (FLRF) models for images. We will discuss the development of several families of RF models defined only over a finite square lattice, say Ω , which is same as that of the given $M \times M$ image to be analysed. The basic idea behind the FLRF model is as follows:

Using a given neighbourhood set N , characterizing the model or system, partition the given image ($M \times M$) lattice into two mutually exclusive subsets Ω_I and Ω_B . The boundary set Ω_B has every grid point which has at least one neighbour defined by N not in Ω . Ω_I is the interior set, equals $\Omega - \Omega_B$. In FLRF models the eqn. for generation

of $y(s)$, $s \in \Omega_I$ and $y(s)$, $s \in \Omega_B$ are different. Specifically every pixel in Ω_I follows the eqn. (4.2) i.e. they are same as that of ILRF model but each pixel in Ω_B follows a modified equation as shown by eqn. (4.13)

$$y(s) = - \sum_{s_k} \beta_k^{(s)} y(s_k) + \sqrt{P} u(s) \quad (4.13)$$

$$s \in \Omega_B, \quad s \in \Omega, \quad s_k \neq s$$

Thus the equation for the image vector \underline{y} can be written as

$$B(\theta) \underline{y} = \sqrt{P} \underline{u} \quad (4.14)$$

where $B(\theta)$ is $M^2 \times M^2$ and \underline{u} is a $M^2 \times 1$ vector of primitive random numbers independent or dependent on the basis of SAR and CM models. By choosing $\beta_k^{(s)}$, $s \in \Omega_B$, appropriately, the matrix $B(\theta)$ can have eigenvectors, either fourier vectors or discrete sine or cosine vectors as the case may be. The distinction between Ω_I and Ω_B can be given as follows

$$\Omega_B = \left\{ s=(i,j), s \in \Omega \text{ and } (s+s_k) \in \Omega \text{ for at least one number } s_k \in N \right\}$$

$$\Omega_I = \left\{ s=(i,j), s \in \Omega \text{ and } (s+s_k) \in \Omega, \forall s_k \in N \right\}$$

The eqn. (4.13) was chosen for the pixels $y(s)$, $s \in \Omega_B$, such that

- i) the equations are closed, i.e. they do not involve any s' where $s' \notin \Omega$
- ii) the overall set of M^2 equations are easy to analyse.

SAR Models:

On the basis of the above mentioned partition of the finite lattice Ω into Ω_I and Ω_B we can write the eqn. (4.2) as

for $s \in \Omega_I$ $y(s)$ obeys (4.2)
but for $s \in \Omega_B$

$$y(s) - \alpha = - \sum_{k \in N} \theta_k [y_1(s+k) - \alpha] + \sqrt{p} u(s) \quad (4.15)$$

where

$$\begin{aligned} y_1(s+k) & \text{ with } s=(i,j) \text{ and } k=(p,q) \\ & = y(s+(p,q)) \text{ if } (s+(p,q)) \in \Omega \\ & = y[(p+i-1)_{M+1}, (q+j-1)_{M+1}] \\ & \text{ if } s+(p,q) \notin \Omega \end{aligned} \quad (4.16)$$

This modified eqn. (4.16) is based on the assumption that the lattice is a torus one, i.e., for such lattices the opposite edges are similar. This is a very useful conception used in this and subsequent chapters.

Eqns. (4.16) and (4.2) give M^2 eqns. relating the image variable $y(\cdot)$ and the IID variable $u(\cdot)$. By using

the same relation as given in eqn. (4.14) we see that for toroidal assumption the matrix $B(\theta)$ becomes a block cyclic of the form,

$$B(\theta) = \begin{bmatrix} B_{11} & B_{12} & \dots & B_{1M} \\ B_{1M} & B_{11} & \dots & B_{1M-1} \\ \vdots & \vdots & \ddots & \vdots \\ B_{12} & \dots & \dots & B_{11} \end{bmatrix}$$

where each B_{ij} is a cyclic submatrix and depends on the parameters $\{\theta_k\}$. For example, when the neighbour set of dependence N is

$$\{(-1,0), (\theta,1), (1,0), (0,-1)\} \text{ then we have,}$$

$$B_{11} = \text{cyc } [1, -\theta_{0,1}, 0, \dots, \theta_{0,-1}]$$

$$B_{12} = \text{cyc } [-\theta_{1,0}, \dots, 0]$$

$$B_{1M} = \text{cyc } [-\theta_{-1,0}, 0, \dots, 0]$$

and $B_{ij} = 0$ for $j \neq 1, 2, M$.

To ensure stationarity, the coefficients θ_k must obey

$$\mu_s(\theta) \triangleq (1 - \theta^T \psi_s) \neq 0, \forall s \in \Omega \quad (4.17)$$

where,

$$\psi_s = \text{col} \left[\exp \left[\sqrt{-1} \frac{2\pi}{M} (s-1)^T r \right], r \in N \right]; \underline{1} = (1, 1)$$

In a similar manner, we can construct the simultaneous ARMA or SARMA model.

Conditional Markov (CM) Model:

In previous chapter we discussed the properties of a CM model defined over an infinite lattice. Again, we can have CM models defined over a finite lattice. We can also tackle the problem of edges by toroidal assumptions. The vector matrix relation in CM model is

$$H(\theta) \underline{y} = \underline{e} \quad (4.18)$$

where $H(\theta)$ is a block cyclic and symmetric. Here to ascertain stationarity we need

$$\mu_s' \triangleq (1 - 2\theta_s^T \phi_s) > 0 \quad \forall s \in \Omega$$

$$\underline{\theta} = \text{col} [\theta_r, r \in N]$$

and

$$\phi_s = \text{col} \left[\cos \frac{2\pi}{M}(s-1)^T r, r \in N_s \right]$$

where N_s is the asymmetrical half of N , i.e.,

$$N = N_s \cup \bar{N}_s$$

$$\bar{N}_s = \left\{ r : -r \in N_s \right\}$$

$$N_s \cap \bar{N}_s = \emptyset$$

4.7 PROPERTIES OF FL MODELS

SAR:

Considering the matrix $B(\theta)$ defined in eqn. (4.14) we can derive some informations about FL SAR model from its toroidal property.

1) Since $B(\theta)$ is a block cyclic matrix, hence the eigenvectors $\mu_{ij}(\theta)$ of $B(\theta)$ are the fourier vectors f_{ij} , where

$$f_{ij} = \text{col} [f_j, \lambda_i f_j \dots \lambda_i^{M-1} f_j] \text{ (Mx1 vector)}$$

and

$$f_j = \text{col} [1, \lambda_j, \lambda_j^2 \dots \lambda_j^{M-1}] \text{ (Mx1 vector)}$$

$$\text{with } \lambda_k = \exp \left[\sqrt{-1} \frac{2\pi k}{M} \right]$$

So given the set of parameters $\{\theta_k\}$ we can write the eqn. for $\mu_{ij}(\theta)$ as

$$\begin{aligned} \mu_{ij}(\theta) &\triangleq A(\lambda_i, \lambda_j) \\ &= 1 + \sum_{(p,q) \in N} \theta_{p,q} \lambda_i^p \lambda_j^q \end{aligned} \quad (4.19)$$

where $k = (p, q)$

2) Again from the vector-matrix relation we can write the covariance Q of Y is as

$$Q = \text{cov} [Y] = [B^T(\theta) B(\theta)]^{-1} \rho$$

$R(k, \ell)$ = autocorrelation at lag (k, ℓ) is given by

$$R(k, \ell) = \frac{1}{M^2} \sum_{i,j=1}^M (\lambda_i^k \lambda_j^\ell) / ||\mu_{ij}(\theta)||^2 \quad (4.20)$$

The discrete spectral density as defined by the DFT (2D) of $R(k, \ell)$ and then for such block cyclic matrix $B(\theta)$ we have

$$S(\gamma_1, \gamma_2) = \frac{1}{||A(e^{\sqrt{-1}\gamma_1}, e^{\sqrt{-1}\gamma_2})||^2} \quad (4.21)$$

$$= \frac{\phi}{||\mu_{ij}(\theta)||^2} \quad \text{for } \gamma_1 = \frac{2\pi i}{M}$$

$$\gamma_2 = \frac{2\pi j}{M}$$

CM Models:

Since in FL toroidal CM model $H(\theta)$ is a block-cyclic symmetric matrix, some properties of CM(FL) models are

1) Similar to FLSAR model, the matrix $H(\theta)$ has fourier vectors as its eigenvectors. If we assume that $\mu_{ij}^1(\theta)$ is the eigenvalue of $H(\theta)$ corresponding to f_{ij} we have

$$\mu_{ij}(\theta) \triangleq G(\lambda_i, \lambda_j)$$

$$= 1 + \sum_{(k, \ell) \in \mathbb{N}} \theta_{k, \ell} \lambda_i^k \lambda_j^\ell$$

2) Without any loss of generality we can assume $\alpha = 0$.

$$\text{Then, } y = [H(\theta)]^{-1} e^{\gamma \theta} \quad (4.22)$$

where \underline{e} is a correlated sequence.

To obtain the covariance of \underline{y} we multiply eqn. (4.22) by \underline{y}^* and take expectations on both sides.

$$\begin{aligned} Q &= [H(\theta)]^{-1} E[\underline{y}^T \underline{e}] \sqrt{\gamma} \\ &= [H(\theta)]^{-1} \gamma \end{aligned} \quad (4.23)$$

from eqn. (4.6).

$$\text{and } \text{cov}[\underline{e}] = E[\underline{e} \underline{e}^T] = H(\theta) \quad (4.24).$$

4.8 RELATIONSHIP BETWEEN SAR AND CM MODELS

In general, two FLRF models defined over same lattice Ω are said to be equivalent if their corresponding output vectors are identical. Unless otherwise stated, equivalence means equivalence in all respects.

Direct comparison of SAR and CM models are hampered by the fact that input \underline{e} to the CM system is correlated. It would be desirable to modify CM model to a new model where the input is uncorrelated like that of SAR. This new representation is also helpful in synthesis of image following a given CM model.

The CM model as given by eqn. (4.22) has the following representation

$$\sqrt{H(\theta)} [\underline{y}] = \sqrt{\gamma} \underline{u} \quad (4.25)$$

$$\sqrt{H(\theta)} = \sum_{i,j=1}^M f_{ij} (f_{ij}^*)^T \sqrt{\mu_{ij}'(\theta)} \quad (4.26)$$

where \underline{u} is the normal zero mean IID sequence and $\sqrt{H(\theta)}$ is real and positive definite.

Eqn. (4.25) again may be written by using eqn. (4.26) as

$$\underline{y} = \left[\sum_{i,j=1}^M f_{ij} (f_{ij}^*)^T \underline{u} / \sqrt{\mu_{ij}'(\theta)} \right] \sqrt{\gamma} \quad (4.27)$$

So \underline{e} in eqn. (4.22) is related to \underline{u} by the eqn.

$$\underline{e} = \sqrt{H(\theta)} \underline{u} \quad (4.28)$$

Even though eqn. (4.25) has a form similar to that of a SAR model but, it is not necessarily a SAR model. A FL CM model represented as

$$G(z_1, z_2) [y(s)] = \sqrt{\gamma} e(s) \quad (4.29)$$

and it is equivalent to a SAR system represented by eqn.

$$B(\theta') \underline{y} = \sqrt{\gamma} \underline{u} \quad (4.30)$$

if $G(z_1, z_2)$ passes a factorization given in eqn. (4.31)

$$\begin{aligned} G(z_1, z_2) &= \delta ||A(z_1, z_2)||^2 \\ &= k A(z_1, z_2) A(z_1^{-1}, z_2^{-1}) \end{aligned} \quad (4.31)$$

If $G(z_1, z_2)$ does not follow eqn. (4.31) but there exist

polynomials $\alpha_1(z_1, z_2)$ and $\alpha_2(z_1, z_2)$ and a constant k such that

$$G(z_1, z_2) ||\alpha_1(z_1, z_2)||^2 = k ||\alpha_2(z_1, z_2)||^2 \quad (4.32)$$

then the corresponding CM model has an equivalent SARMA model. If $G(z_1, z_2)$ is not at all factorizable then the CM model has no equivalent simultaneous model.

4.9 PARAMETER ESTIMATION

Suppose we have already specified the model type and the neighbour set. Let $\underline{\phi} = (\underline{\theta}, \rho)$ be the vector of unknown parameters, where $\underline{\theta}$ is the vector of parameters and ρ is the noise variance. The problem of estimating $\underline{\phi}$ from the given image vector \underline{y} is primarily a statistical problem. This problem can be solved either by least square (LS) method or maximum likelihood (ML) method. In the case of LS estimate the estimates are given by

$$\hat{\underline{\theta}} = [\sum z(s) z^T(s)]^{-1} (\sum z(s) y(s)) \quad (4.33)$$

and

$$\hat{\rho} = \frac{1}{M^2} \sum (y(s) - \hat{\underline{\theta}} z(s))^2 \quad (4.34)$$

where

$$z(s) = \text{col } [y(s+r), r \in N] \quad (4.35)$$

It has been shown by Ord [37] that this method does not give consistent result for nonunilateral neighbourhood.

Maximum Likelihood Estimate (MLE):

In case of FL images, the MLE gives a consistent and efficient estimates of the parameters. For such MLE we have to find the log likelihood function of the output vector \underline{y} . To obtain such an expression we impose the condition that input IID sequence is a zero-mean Gaussian which gives a Gaussian output as the system is linear. If we represent the system by its transfer function $A(z_1, z_2)$ or by its matrix form $B(\theta)$ then likelihood function comes out to be

$$p(\underline{y} | \underline{\theta}, \rho) = J \exp\left[-\frac{1}{2\rho} \sum_{s \in \Omega} A(z_1, z_2) y(s)\right]^2 \quad (4.36)$$

where J is the Jacobian

or,

$$\ln p(\underline{y} | \underline{\theta}, \rho) = \ln [\det B(\theta)] - \frac{M^2}{2} \ln (2\pi \rho) - \frac{1}{2\rho} \sum_{s \in \Omega} (y(s) - \theta^T z(s))^2 \quad (4.37)$$

$$\text{since } \det [B(\theta)] = \prod_{s \in \Omega} (1 - \theta^T \psi_s)$$

we have

$$\ln p(\underline{y} | \underline{\theta}, \rho) = \sum_{s \in \Omega} \ln(1 - \theta^T \psi_s) - \frac{M^2}{2} \ln(2\pi \rho) - \frac{1}{2\rho} \sum_{s \in \Omega} (y(s) - \theta^T z(s))^2 \quad (4.38)$$

The ML estimates are obtained by maximising eqn. (4.35) w.r.t. $\underline{\theta}$ and ϕ . Since log likelihood function is a quadratic in $\underline{\theta}$, the estimation may be done by numerical optimization schemes like Newton-Raphson etc., which are computationally expensive.

Here we shall describe an iterative scheme for estimation which is an approximate version of the MLE. In such a scheme we approximate expressions like $\ln(1+a)$ by $a - a^2/2$. The corresponding log likelihood function denoted by $J(\underline{\theta}, \phi)$ becomes

$$J(\underline{\theta}, \phi) = -\underline{V}^T \underline{\theta} + 0.5 \underline{\theta}^T R \underline{\theta} - \frac{M^2}{2} \ln(2\pi\phi) - \frac{1}{2\phi} \sum_{s \in \Omega} (y(s) - \underline{\theta}^T \underline{z}(s))^2 \quad (4.39)$$

where,

$$\underline{V} = \sum_{s \in \Omega} C_s, \text{ } m \times 1 \text{ vector} \quad (4.40)$$

$$R = \sum_{s \in \Omega} (S_s S_s^T - C_s C_s^T), \text{ } mxm \text{ matrix} \quad (4.41)$$

$$\underline{C}_s = \text{col} \left[\cos \frac{2\pi}{M} ((s-1)^T \underline{r}), r \in N \right] \quad (4.42)$$

$$\underline{S}_s = \text{col} \left[\sin \frac{2\pi}{M} ((s-1)^T \underline{r}), r \in N \right] \quad (4.43)$$

After defining the matrices and vectors the algorithm for approximate MLE is as follows:

$$\underline{\theta}_{t+1} = (R - \frac{1}{f_t} \underline{S})^{-1} (\underline{V} - \frac{1}{f_t} \underline{U}) \quad (4.44)$$

and

$$f_t = \frac{1}{M^2} \sum_{s \in \Omega} (y(s) - \underline{\theta}_t^T z(s))^2 \quad (4.45)$$

where

$$\underline{S} = \sum_{s \in \Omega} z(s) z(s)^T \quad (4.46)$$

(mxm matrix)

$$\underline{U} = \sum_{s \in \Omega} z(s) y(s) \quad (4.47)$$

(mx1 vector)

The initial value $\underline{\theta}_0$ is chosen as $\underline{\theta}_0 = S^{-1} \underline{U}$, and here m is the dimension of $\underline{\theta}$.

CM Model:

An estimate with good consistency and efficiency may be obtained in CM models by ML estimates. But it is not easy to derive an explicit expression for the log likelihood function for such CM models because Jacobian of the transformation matrix is difficult to be evaluated. By assuming a toroidal structure and Gaussian correlated sequence $e(s)$, the log-likelihood function may be given as

$$\ln p(y | \underline{\theta}, \gamma) = \sum \frac{1}{2} \ln (1 - 2\underline{\theta}^T \underline{q}_s) - \frac{M^2}{2} \ln(2\pi \gamma) - \frac{1}{2\gamma} \underline{y}^T H(\underline{\theta}) \underline{y} \quad (4.48)$$

We see that the exponential term in $\ln p(y | \underline{\theta}, \gamma)$ is linear in $\underline{\theta}$ unlike that of SAR model. A computationally efficient and consistent estimation procedure is as follows

$$\underline{\theta}^* = \left[\sum_{s \in \Omega_I} q(s)q(s)^T \right]^{-1} \left(\sum_{s \in \Omega_I} q(s)y(s) \right) \quad (4.49)$$

and

$$\gamma^* = \frac{1}{M^2} \sum_{s \in \Omega_I} (y(s) - \underline{\theta}^{*T} q(s))^2 \quad (4.50)$$

where Ω_I is the interior region. This procedure is suggested by Woods [36]. In our study, estimation of parameters for both SAR and CM followed the above mentioned scheme and experimental results show that the estimates are quite close to the actual values and moreover this scheme works well for non Gaussian cases also (see section 4.11).

4.10 CHOICE OF NEIGHBOURHOOD FOR SAR AND CM MODELS

From 1D time series modelling, it is known that only a model of appropriate ^{order} \angle gives the best result for a set of discrete data for forecasting and prediction. A similar situation is true in 2D spatial processes, but, the

problem with 2D data set is the variety of models available. So if we want to reconstruct any image from the parameters extracted from it, then the quality of reconstructed image depends on how the underlying model is similar to the true model and here comes the importance of the choice of appropriate neighbourhood.

Though, one way of picking up the appropriate model is by visual inspection, we give below a quantitative measure for this. One way is to do pairwise hypothesis testing. But such scheme is not consistent and also not transitive. If a model C_1 is preferred to C_2 and C_2 is preferred to C_3 then it does not follow C_1 is preferred to C_3 . Such a decision rule is not consistent in the sense that the probability of choosing an incorrect model does not go to zero if we take infinite number of observations. The other choice is based on Bayesian decision rule [35].

Bayes Rule for Choosing SAR Model:

Suppose that we have three sets N_1 , N_2 and N_3 of neighbours having m_1, m_2 and m_3 members respectively. For each model N_k , we can assign one SAR model C_k

$$y(s) = \sum_{r \in N_k} \theta_k y(s+r) + \sqrt{\rho_k} u(s) \quad s \in \Omega_I$$

$$y(s) = \sum_{r \in N_k} \theta_k y_I(s+r) + \sqrt{\rho_k} u(s) \quad s \in \Omega_B$$

where the significance of such equations are already defined.

The models C_i , $i = 1, 2, 3$, are mutually exclusive. By Bayesian rule we can say that the optimal decision rule for minimising the average probability of decision error is to choose the model C_i that maximises the posterior probability $P(C_i | \underline{y})$, where \underline{y} is the observation vector. The quantity $P(C_i | \underline{y})$ is computed by Bayesian rule, $P(C_i | \underline{y}) = P(\underline{y} | C_i) P(C_i) / P(\underline{y})$. We can set $P(C_i)$ same for all i in the absence of any contrary information, so that

$$p(\underline{y} | C_k) = \int p(\underline{y} | \underline{\theta}, \phi) p(\underline{\theta}, \phi | C_k) d\underline{\theta} d\phi$$

This expression, with the Gaussian assumption is similar to the expression which is obtained in the section on parameter estimation. By assuming that $p(\underline{\theta}, \phi | C_i)$ is a-priori density then by using asymptotic information it can be shown that

$$p(\underline{y} | C_k) \approx M^2 / 2 \ln \bar{P}_k + \frac{1}{2} \sum_{s \in \mathcal{S}} \ln(1 - \underline{\theta}_k^{-T} C_{ks} + \underline{\theta}_k^{-T} Q_{ks} \bar{\theta}_k) - \frac{m_k}{2} \ln(M^2) + \ln p(\bar{\theta}_k, \phi_k | C_k).$$

In general, computation of such an expression for all values of k is not computationally efficient so here we will give an approximate test statistic by dropping the terms due to prior densities. Then the decision rule for the choice of appropriate neighbour set is following.

Choose the neighbour set N_k if

$$k = \left| \min_k \{C_k\} \right| \quad (4.51)$$

where

$$C_k = \left\{ - \sum \ln(1 - 2 \theta_k^{-T} C_{ks} + \theta_k^{-T} Q_{ks} \bar{\theta}_k) + M^2 \ln \bar{\phi}_k + m_k \ln(M^2) \right\}$$

$$Q_{ks} = S_{ks} S_{ks}^T + C_{ks} C_{ks}^T$$

$$C_{ks} = \text{col} \left[\cos \frac{2\pi}{M} ((s-1)^T r), r \in N \right]$$

$$S_{ks} = \text{col} \left[\sin \frac{2\pi}{M} ((s-1)^T r), r \in N \right]$$

Bayes Rule for Choosing CM Model:

Suppose, we have three sets N_1, N_2 and N_3 of neighbours having $2m_1, 2m_2$ and $2m_3$ members (symmetric) respectively. For any model N_k , we can write,

$$\begin{aligned} y(s) &= \sum_{r \in N_k} \theta_{kr} y(s+r) + \sqrt{\phi}_k e(s), s \in \mathcal{R}_I \\ &= \sum_{r \in N_k} \theta_{kr} y_1(s+r) + \sqrt{\phi}_k e(s), s \in \mathcal{R}_B \end{aligned}$$

where the symbols are defined previously, and $e(s)$ is correlated Gaussian sequence. Then in the similar way, of SAR, we can say that the approximate test statistics will be:

Choose the set N_k if

$$k = \left| \min_k \left\{ C_k \right\} \right| \quad (4.52)$$

where,

$$C_k = - \sum_{s \in N} \ln (1 - \hat{\theta}_k^T \phi_{ks}) + M^2 \ln(\gamma_k) + m_k \ln(M^2)$$

where,

$$\hat{\theta}_k = \text{col} [\hat{\theta}_k, k \in N_{s_k}]$$

and

$$\phi_{ks} = \text{col} \left[\cos \frac{2\pi}{M} (s-1)^T r \right], r \in N_{s_k}$$

where N_{s_k} is the asymmetrical half of N_k , i.e.,

$$N_k = N_{s_k} \cup \bar{N}_{s_k}$$

$$\bar{N}_{s_k} = \left\{ r : -r \in N_{s_k} \right\}, N_{s_k} \cap \bar{N}_{s_k} = \emptyset$$

In the next section where we will discuss the simulation results, we will show how test statistics detects the actual model for synthetic images for both SAR and CM models.

4.11 EXPERIMENTAL RESULTS

To see how the SAR and CM models and associated estimation scheme work, different images, using different models have been generated and displayed for visual tests.

The fast schemes for generating toroidal synthetic images by a known SAR and cal models are given below.

Generation scheme for MxM image (SAR):

i) Generate a sequence of IID numbers (Gaussian or uniform) of length M^2 and arrange them to form a MxM array. Let that array be U .

ii) Given a set of neighbours θ_r find the eigenvalue matrix μ of the system as

$$\mu_{ij}(\theta) = 1 + \sum_{r \in N} \theta_r z_i^k z_j^l \quad \text{where } r=(k,l)$$

iii) Take the 2D D.F.T. of the array U . Let the transformed array be F_U

iv) Generate an array F_Y where

$$F_Y(i,j) = (F_U(i,j)/\mu(i,j)) \sqrt{\rho} \quad (4.53)$$

v) The final image obeying the given SAR model is

$$\begin{aligned} Y &= \frac{1}{M^2} F^{-1}(F_Y) \\ &= \frac{1}{M^2} F^{-1}[(F_U(i,j)/\mu(i,j)) \sqrt{\rho}] \end{aligned}$$

In actual practice we did all the transforms by the IMSL routine FFT2 and FFRDR2.

The above algorithm for generating image was used with different neighbourhood (pre-specified) and with IID

sequences with different densities. Parameters were estimated from the generated image by using approximate ML estimates. In any random number generation, we require some initial values (called ISEED in IMSL) and it has been found by experiments that estimated parameters change with different ISEED value and that change is also quite random.

Generation Scheme for MxM Image (CM):

Since the generating eqn. in CM model can be written as

$$\sqrt{H(\theta)} y = \sqrt{\mu} u$$

hence the generating strategy for CM is same as that of SAR except the eqn. (4.53) where in CM, eqn. (4.53) is modified to

$$F_y = (F_U(i,j) / \sqrt{\mu(i,j)}) \sqrt{\mu} \quad (4.54)$$

Hence also we take uniform as well as Gaussian density for the IID sequence .

Results for SAR and CM model parameter estimations for different ISEED value will be given in tabular form.

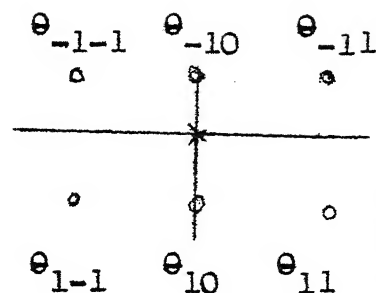
In the next set of tables we will give the estimated parameters with the test statistics C_k for finding the optimal neighbourhood. Here we have considered only noncausal neighbourhood with 2,4,6 and 8 parameters for both SAR and CM models.

GAUSSIAN SAR ESTIMATED

ACTUAL

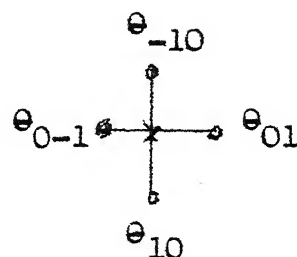
(6 Point)

| ρ | =1.1111 | 0.9982 | 1.035530 | 1.081630 |
|------------|---------|---------|----------|----------|
| e_{-11} | 0.28 | 0.2988 | 0.3012 | 0.2877 |
| e_{10} | 0.12 | 0.1372 | 0.1464 | 0.1220 |
| e_{-1-1} | -0.14 | 0.1612 | -0.1589 | -0.1729 |
| e_{11} | -0.14 | -0.1612 | -0.1589 | -0.1729 |
| e_{10} | 0.12 | 0.1372 | 0.1464 | 0.1220 |
| e_{1-1} | 0.28 | 0.2988 | 0.3012 | 0.2877 |



GAUSSIAN SAR (4 POINT)

| ρ | =1.1111 | 1.02123 | 1.07213 | 1.00213 |
|-----------|---------|---------|---------|---------|
| e_{10} | 0.36 | 0.4123 | 0.4005 | 0.3823 |
| e_{0-1} | -0.12 | -0.1423 | -0.1252 | -0.1512 |
| e_{01} | -0.12 | -0.1423 | -0.1252 | -0.1512 |
| e_{10} | 0.36 | -0.1423 | 0.4005 | 0.3823 |



UNIFORM (SAR) 6 POINT

| ρ | = 1.1111 | 1.021346 | 0.998312 | 1.021672 | | | |
|--------|----------|----------|----------|----------|------------|-----------|-----------|
| | 0.28 | 0.3087 | 0.3054 | 0.3124 | | | |
| | 0.12 | 0.1423 | 0.1482 | 0.1480 | e_{-1-1} | e_{-10} | e_{-11} |
| | -0.14 | -0.1642 | -0.1601 | -0.1650 | | | |
| | -0.14 | -0.1642 | -0.1601 | -0.1650 | | | |
| | 0.12 | 0.1423 | 0.1482 | 0.1430 | e_{1-1} | e_{10} | e_{11} |
| | 0.28 | 0.3087 | 0.3854 | 0.3124 | | | |

CM (6 POINT) GAUSS

| ρ | = 1.1111 | 1.2312 | 1.01245 | 0.99824 | | | |
|------------|----------|---------|---------|---------|-----------|-----------|----------|
| e_{-11} | .2800 | 0.2912 | 0.2992 | 0.3012 | | e_{-10} | |
| e_{-10} | 0.1200 | 0.1421 | 0.1291 | 0.1282 | e_{0-1} | | e_{01} |
| e_{-1-1} | -0.1400 | -0.1232 | -0.1501 | -0.1542 | | e_{10} | |
| e_{11} | -0.1400 | -0.1232 | -0.1501 | -0.1542 | | | |
| e_{10} | 0.1200 | 0.1421 | 0.1291 | 0.1282 | | | |
| e_{1-1} | 0.2800 | 0.2912 | 0.2992 | 0.3012 | | | |

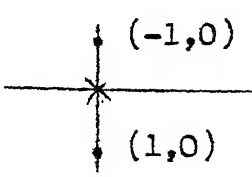
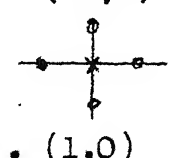
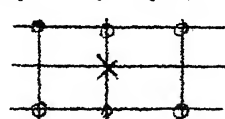
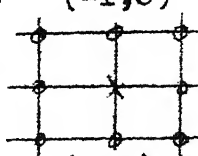
CM (4 POINT)GAUSS

| | ACTUAL | ESTIMATED | | | | | |
|------------|--------|-----------|---------|---------|------------|--|-----------|
| e_{-1-1} | 0.28 | 0.2942 | 0.2852 | 0.2915 | | | |
| e_{-11} | -0.14 | -0.1240 | -0.1493 | -0.1413 | e_{-1-1} | | e_{-11} |
| e_{1-1} | -0.14 | -0.1240 | -0.1493 | -0.1413 | | | |
| e_{11} | 0.28 | 0.2942 | 0.2852 | 0.2915 | | | |
| ρ | 1.1111 | 1.0243 | 1.0823 | 1.1056 | e_{1-1} | | e_{11} |

CM (4 POINT) UNIFORM

| | | | | | | |
|--------|---------|---------|---------|------------|--|-----------|
| 0.28 | 0.2752 | 0.2901 | 0.2713 | e_{-1-1} | | e_{-11} |
| -0.14 | -0.1482 | -0.1391 | -0.1542 | | | |
| -0.14 | -0.1482 | -0.1391 | -0.1542 | | | |
| 0.28 | 0.2752 | 0.2901 | 0.2713 | e_{1-1} | | e_{11} |
| 1.1111 | 1.1623 | 1.1192 | 1.2109 | | | |

Computation of Parameter C_K for an Image (following SAR Model) when the real neighbourhood is $N = (0,1), (0,-1), (1,0), (-1,0)$. Original $\theta_{10} = \theta_{-10} = 0.36$; $\theta_{01} = \theta_{0,-1} = -0.12$; $\phi = 1.1111$

| | θ | C_K | Positions |
|--------|--|--------|--|
| 1.0355 | 0.4858 0.4858 | 179.42 |  |
| 1.0052 | 0.4005 -0.1152 -0.1152 0.4005 | 80.76* |  |
| 1.0686 | 0.5616×10^{-1} 0.4041 0.5550×10^{-1} 0.5550×10^{-1} 0.4041 0.5616×10^{-1} | 587.48 |  |
| 1.0143 | 0.6963×10^{-2} 0.3966 0.7091×10^{-2} -0.1067 -0.1067 0.7091×10^{-2} 0.3966 0.6963×10^{-2} | 230.07 |  |

Computation of the parameter C_k for an image (following CM model) when the real neighbourhood is $N = (1,1), (-1,-1), (-1,1)$. Original $\theta_{-11} = \theta_{1-1} = -0.14$ $\rho = 1.1111$
 $\theta_{11} = \theta_{-1-1} = 0.28$

| | θ | C_k | Positions | |
|--------|---|--------|-------------|-----------|
| 1.1292 | $\theta_{11}=0.3221$ | 420.72 | -1-1 | 11 |
| 1.4213 | $\theta_{-11}=0.2121$ | 621.63 | -11 | 1-1 |
| 1.0321 | $\theta_{-11}=0.1512$ $\theta_{11}=0.2922$ | 121.43 | -1-1 1-1 | -11 11 |
| 1.1021 | $\theta_{10}=0.0212$ $\theta_{01}=0.0092$ | 412.76 | -10 0-1 | 01 |
| | | | | 10 |

Real Image Synthesis using SAR Models:

Our discussions so far have been concerned with synthetic image patterns generated by known SAR models. We conclude this chapter by studying the appropriateness of the SAR models to real images. For this purpose, we have taken a 32x32 portion of Monalisa image. Since the entire image may not be homogeneous we have divided the entire image into 4 blocks of size 16x16. Parameters ($\underline{\theta}, \underline{f}$) were estimated for each block by the approximate ML estimate scheme with 12 point neighbourhood. The neighbourhood includes the points (1,0), (1,1), (0,1), (-1,1), (-1,0), (-1,-1), (0,-1), (1,-1), (2,0), (0,2), (-2,0), (0,-2). We have also estimated the parameters considering 8 point neighbourhoods. In that case, the neighbourhood N contains (1,0), (1,1), (0,1), (-1,1), (-1,0), (-1,-1), (0,-1), (1,-1). In the case of 8 point neighbourhood the quality of the synthesized image was not good. From the estimated parameters and sample means, we compute the residuals ω as

$$\omega = \frac{1}{S} [H(\underline{\theta}) \underline{y} - \alpha \underline{1}]$$

where \underline{y} is the block image and α is the sample mean. From this sequence $\underline{\omega}$ we have generated a set of random numbers whose histogram was approximately same as the histograms of the residuals. The new sequence and the parameter

set were then used to synthesize the image. The synthesized image closely resembles the actual one except the edge effect of the block operation.

4.12 DISCUSSION

In this chapter we have given a complete methodology for analysing a given set of spatial data. We have also discussed the estimation scheme for SAR and CM models from the computation of test statistics. We have not said any thing about the superiority of SAR over CM model or vice-versa as the appropriateness of one class of model is entirely dependent on the data itself.

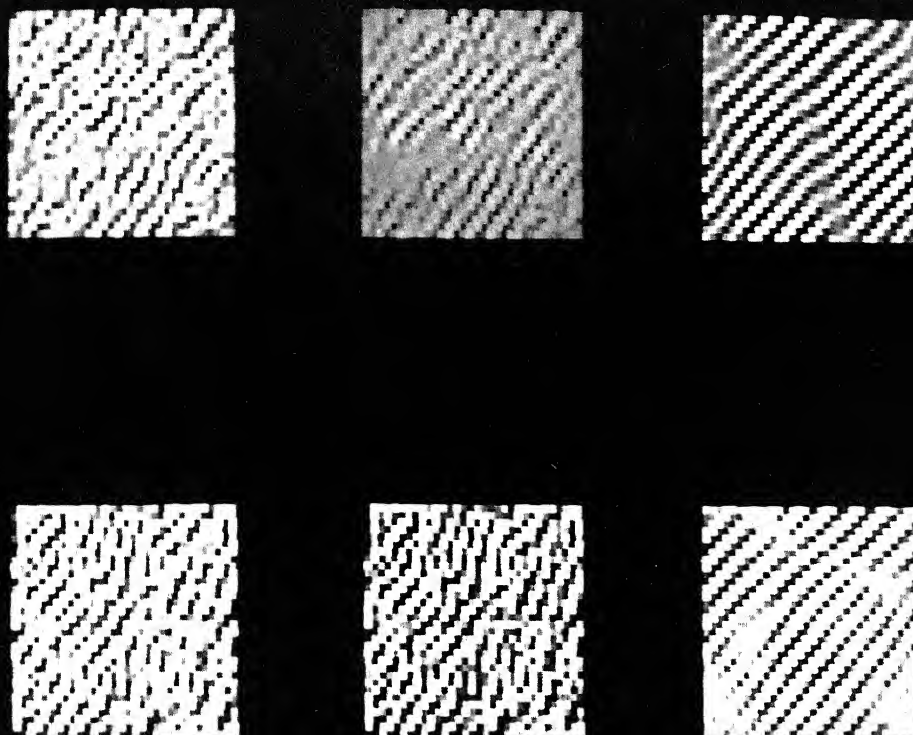


Fig. 4.1: (1,1): original (6 point); estimated image with 6 point and (SAR) 4 point are shown in (1,2) and (1,3) respectively for Gauss (2,1) to (2,3) are same as (1,1) to (1,3) with uniform density.

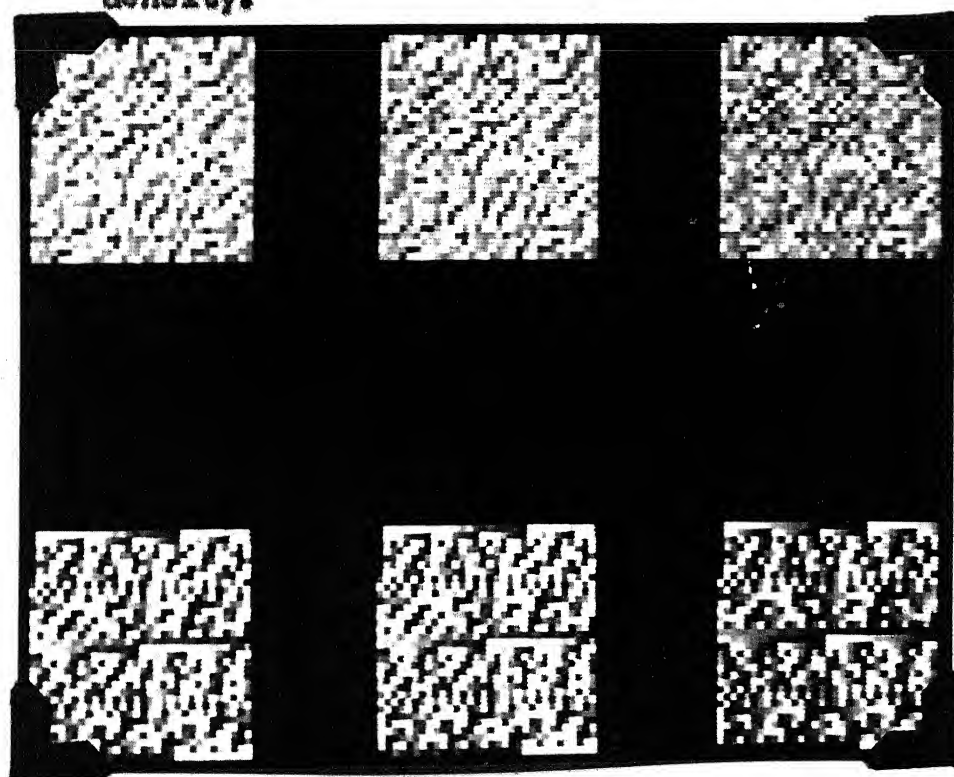


Fig. 4.2: (1,1): original (6 point); estimated images with 6 point and (CM) 4 point are shown in (1,2) and (1,3) respectively for Gauss (2,1) to (2,3) are same as (1,1) to (1,3) with uniform density.

CHAPTER 5

TEXTURE SEGMENTATION BY SAR MODEL

5.1 PRINCIPLE OF SEGMENTATION

Segmentation is a process that breaks up a composite scene into its constituents. Though there are several approaches for tackling this problem [3] yet it is an active field of research because of its importance as the first processing step in any practical scene analysis problem. In natural composite textures (e.g. combination of land and sea) we may or may not have any sharp boundary as the line of demarcation. In such cases where the contour is missing, it has been observed that statistical measure gives an estimated contour which is very close to the original one. Here segmentation by gray level thresholding will not work satisfactorily because there may be two regions having the same brightness average but slightly different variances so that it is not distinguishable. In a composite image (texture) having two homogeneous regions we assume that all the pixels in one region are having same statistical properties. In fact, a composite texture may have more than two homogeneous regions but solving a two region problem can always be generalized for n-region textures.

Though there are several statistical segmentation schemes, the proposed scheme for segmenting a random field composite texture based on spatial interaction model is superior to them in the sense that it is computationally very simple and to some extent accurate. In this chapter we will discuss the principle of such segmentation scheme and the associated results.

5.2 APPLICATION OF RF MODEL FOR SEGMENTATION

Given a composite texture X , we assume that for each homogeneous part of it we can have a RF model and model parameters are different from one region to another. Let

$$X = X_1 \cup X_2 \quad (5.1)$$

where X_1 and X_2 are the homogeneous random textures. Since X_1 and X_2 are following a toroidal SAR model, we write

$$X_1 : y_1(s) + \sum_{k_1 \in N_1} \theta_{k_1} y_1(s+k_1) = \sqrt{\rho_1} u(s), s \in \Omega_I \quad (5.2)$$

$$y_1(s) + \sum_{k_1 \in N_1} \theta_{k_1} y_1'(s+k_1) = \sqrt{\rho_1} u(s) \quad s \in \Omega_B$$

similarly,

$$X_2 : y_2(s) + \sum_{k_2 \in N_2} \theta_{k_2} y_2(s+k_2) = \sqrt{\rho_2} u(s) \quad s \in \Omega_I \quad (5.3)$$

$$y_2(s) + \sum_{k_2 \in N_2} \theta_{k_2} y_2'(s+k_2) = \sqrt{\rho_2} u(s) \quad s \in \Omega_B$$

where the various parameters appearing in eqn. (5.2) and eqn. (5.3) are explained in the previous chapter. The entire image is composed of two regions; region-1 and region-2, represented by $\{\underline{\theta}_1, \rho_1\}$ and $\{\underline{\theta}_2, \rho_2\}$ respectively, where $\underline{\theta}_1 = \{\theta_{k_1}\}$ and $\underline{\theta}_2 = \{\theta_{k_2}\}$. After estimating the parameters for each region we use them to find the similarity of them inside a region. The segmentation procedure begins by breaking the entire image into smaller square blocks. We then consider each block image as an independent one and the parameters for each block is estimated. The same procedure is followed for all such blocks. We thus obtain a set of parameter for each block. The final step is to cluster these set of data by any standard scheme. The clustering of these data will determine the segmenting line. Though the entire image may be correlated, for the purpose of segmentation, we assume the block images to be independent. This is not a valid assumption and will give some error in the final result. But experience has shown that this error is negligible with moderate block size (8x8) and it varies inversely with the window size. The entire process can be visualized by Fig. 5.1.

Here we consider a $M \times M$ image with windows of size $N \times N$. So, effectively, we will have K such blocks where

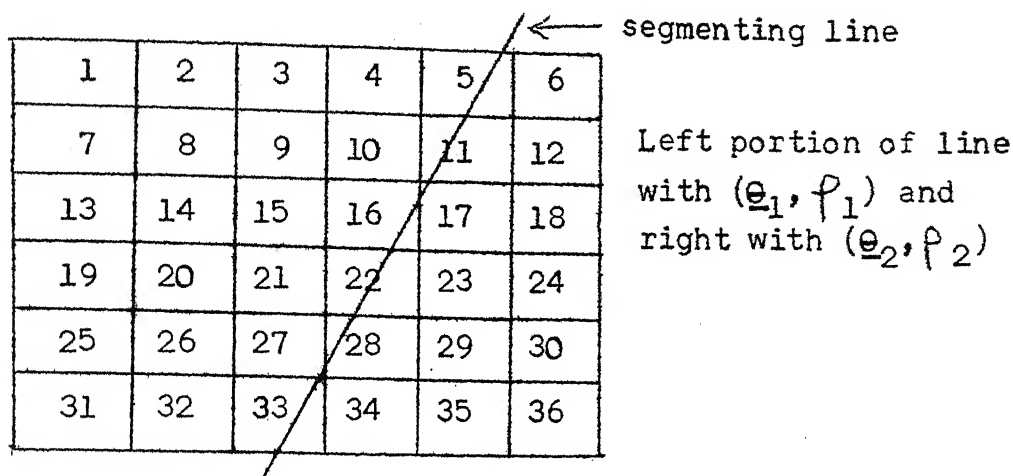


Fig. 5.1: A Block Composite Image

$K = M^2/N^2$. We estimate the parameters of each block, treating them as independent, and arrive at K sets of $\{e, p\}$. Then by clustering these parameters we can separate the non-uniform blocks. Since clustering in 2 region problem is a binary decision making process, the estimated curve will always be a staircase one. Evidently, the estimated curve will be close to the actual one if the block size is reduced but, we will show that it has other implications also. This method is tested with different segmenting curves like straight lines, parabolas, circles and for each case satisfactory results are obtained.

5.3 PRINCIPLE OF CLUSTERING

In this section we will discuss the principle of clustering and the algorithm used. Clustering is also known as unsupervised learning. Roughly speaking, clustering

procedures yield a data description in terms of clusters or groups of data points that possess strong internal similarities. There are several methods of clustering for a set of data but all are heuristic in the sense that they do not possess any mathematical properties but are easy to use [38]. Since the clustering procedures have no mathematical foundations, they depend on the nature of the data. For all algorithms at least some properties of the data should be known a priori.

In this study, the algorithm used for the segmentation procedure, is known as K-mean algorithm [39]. In this algorithm we assume that the number of clusters at final stage is K . This is one approach of general iterative optimization scheme. We will describe the algorithm and also give an example of it.

The K-mean algorithm is based on the minimization of performance index which is defined as the sum of square errors. This algorithm consists of the following steps.

1. Choose K initial cluster centres $z_1(1), z_2(1), \dots, z_K(1)$. These are arbitrary and are usually selected as the first K samples of the given sample set.
2. At the K th iterative step, distribute the samples $\{X\}$ among the K cluster domain using the relation

$x \in S_j(K)$ if

$$||x - z_j(K)|| < ||x - z_i(K)||, \forall i=1,2,\dots,K, i \neq j \quad (5.4)$$

where $S_j(K)$ denotes the set of samples whose cluster centre is $Z_j(K)$. Ties in expression (5.4) are solved arbitrarily.

3. Compute the new cluster centre $Z_j(K+1)$, $\forall j=1,2,\dots,K$ from the result of step 2 such that the sum of square distance from all points in $S_j(K)$ to the new cluster centre is minimised. In other words, the new cluster centre $Z_j(K+1)$ is computed so as to minimise the performance index J_j , where,

$$J_j \triangleq \sum_{x \in S_j(K)} ||x - Z_j(K+1)||^2 \quad j=1,2,\dots,K \quad (5.5)$$

The $Z_j(K+1)$ which minimises J_j is simply the sample mean $S_j(K)$. So, the new cluster centre is given by

$$Z_j(K+1) = \frac{1}{N_j} \sum_{x \in S_j(K)} x, \quad j=1,2,\dots,K \quad (5.6)$$

where N_j is the number of samples in $S_j(K)$. The name 'K-mean' is obviously derived from the manner in which the centres are sequentially updated.

4. If $Z_j(K+1) = Z_j(K)$ for $j = 1,2,\dots,K$,

the algorithm has converged and the procedure has terminated. Otherwise go to step 2.

Convergence of such a scheme is not proved and convergence depends on the value of K and the type of data.

We will also give an example of such scheme for a 2D data.

Example:

1. Let $K=2$ and let us take

$$Z_1(1) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = X_1$$

and

$$Z_2(1) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = X_2$$

2. Since

$$||X_1 - Z_1(1)|| < ||X_1 - Z_2(1)||$$

and

$$||X_3 - Z_1(1)|| < ||X_3 - Z_2(1)||$$

So,

$$S_1(1) = \{X_1, X_3\} \quad i=2$$

$$S_2(1) = \{X_2, X_4, \dots, X_{20}\}$$

3. Update the cluster centres

$$Z_1(2) = \frac{1}{N_1} \sum_{x \in S_1(1)} x = \frac{1}{2}(X_1 + X_3) = \begin{bmatrix} 0.00 \\ 0.50 \end{bmatrix}$$

$$Z_2(2) = \frac{1}{N_2} \sum_{x \in S_2(1)} x = \frac{1}{18}(X_2 + X_4 + \dots + X_{20}) = \begin{bmatrix} 5.67 \\ 5.63 \end{bmatrix}$$

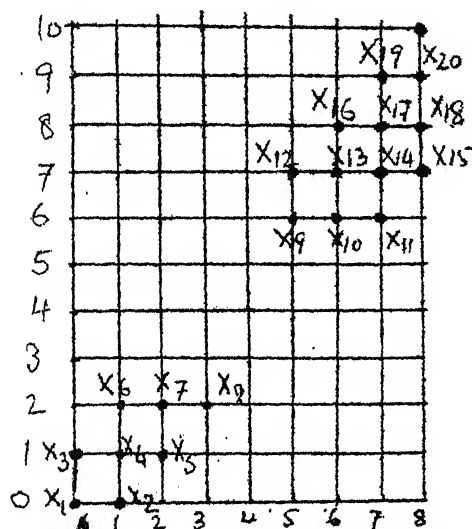


Fig. 5.2: 2D Clustering

4. Since $Z_j(2) \neq Z_j(1), \forall j$, so go to (2)

2. With new cluster centres we get,

$$||x_\ell - Z_1(2)|| < ||x_\ell - Z_2(2)||, \forall \ell = 1, 2, \dots, 8$$

and

$$||x_\ell - Z_2(2)|| < ||x_\ell - Z_1(2)||, \forall \ell = 9, \dots, 20$$

$$S_1(2) = \{x_1 \dots x_8\}$$

$$S_2(2) = \{x_9 \dots x_{20}\}$$

3. Update the cluster centre

$$Z_1(3) = \frac{1}{8} (x_1 + \dots + x_8) = \begin{bmatrix} 1.25 \\ 1.13 \end{bmatrix}$$

$$Z_2(3) = \frac{1}{12} (x_9 + \dots + x_{20}) = \begin{bmatrix} 7.67 \\ 7.33 \end{bmatrix}$$

4. Since $Z_j(3) \neq Z_j(2), j=1,2$ so go to step 2. Step 2 yields same result as in last iteration so $Z_1(4)=Z_1(3)$ and $Z_2(4) = Z_2(3)$. Step 3 also yields the same results. Since $Z_j(4) = Z_j(3), j=1,2$ the algorithm has converged and it has yielded the cluster centres

$$Z_1 = \begin{bmatrix} 1.25 \\ 1.13 \end{bmatrix} \text{ and } Z_2 = \begin{bmatrix} 7.67 \\ 7.33 \end{bmatrix}.$$

This algorithm is very sensitive to the initial cluster centres but generally the first two data in the data set are taken as the cluster centres for the first iteration.

Though the above example deals with clustering of 2 dimensional vector but in practice for image segmentation we need clustering of feature vectors of dimension $n+1$ (for n -point neighbourhood).

5.4 SIMULATION RESULTS

In this section we will discuss the simulation results obtained by the above scheme. Various composite textures were generated using different SAR models. We have taken only 64×64 image but various window sizes were used for segmentation purposes. An empirical relation between the window size and error is given where error is defined in the mean square sense. If $\underline{\theta}_\omega$ represents the parameter vector for a fixed window ω and $\hat{\underline{\theta}}_\omega$ be its estimate then the error is defined as

$$\begin{aligned} \epsilon_\omega &= | \underline{\theta}_\omega - \hat{\underline{\theta}}_\omega | \\ \text{or,} \quad \epsilon_\omega &= \sqrt{\frac{1}{N} \sum_{i=1}^N (\theta_{\omega_i} - \hat{\theta}_{\omega_i})^2} \end{aligned} \quad (5.7)$$

where N is the length of vector $\underline{\theta}$.

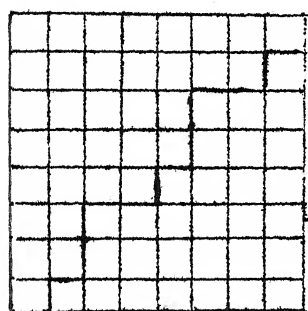
In the following, we present the method adopted for the segmentation scheme and the method of error estimation for the synthetic composite texture. The generated image has two parts or regions as shown in the Fig. 5.3.

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 |
| 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |

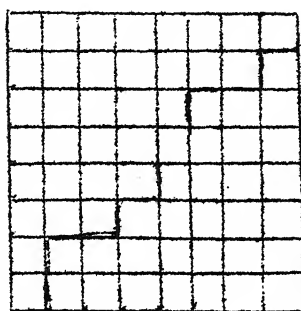
Fig. 5.3: Composite Texture

This image was actually generated by superposition of two different images such that, in the composite image left portion of the line (at 45°) has parameter $(\underline{e}_1, \rho_1)$ and the right portion is with parameter $(\underline{e}_2, \rho_2)$. In the actual simulation $\underline{e}_1 = [0.12 \ 0.12]^T$ and $\rho_1 = 1.1111$ and $\underline{e}_2 = [0.36 \ 0.36]^T$ and $\rho_2 = 1.9999$ was used. We partitioned the entire 64×64 image into blocks of sizes 8×8 and 4×4 . In case of 8×8 blocks we get 64 such blocks and 256 in case of 4×4 blocks. In Fig. 5.3 the line passes through the blocks 8, 15, 22, 29, 36, 43, 50 and 57 and they are really the composite blocks. Parameters were estimated for each blocks and all the blocks on the left side of the line yields nearly same parameters as is to be expected and same is true for the blocks at the right side. After parameter estimation,

we have a set of 64 (for 8×8) vectors where each vector has three components when we estimate a 2 point problem. These vectors are called feature vectors and they are used for clustering. Since we have considered symmetric neighbourhood, the 3 point vectors reduces to 2 point because of the symmetry. These vectors then can be plotted on a 2D plane. Graphically, the separations are distinguishable. As mentioned earlier, all clustering schemes are heuristic and they have inherent limitations in separating data points. In the present case also, we end up with a few misclassified samples but they are easily detectable. We show a segmentation result for the same texture for 2 point and 4 point parameter estimation in Figure 5.4.



2 Point



4 Point

Fig. 5.4: Segmenting line estimated by different SAR models.

In Fig. 5.5, we have given the actual and estimated segmenting curves. In order to study the error in estimated parameter with window size, we have taken a pure homogeneous image having $\underline{\theta} = \begin{bmatrix} 0.12 \\ 0.12 \end{bmatrix}$ and $\phi = 1.1111$. Then the parameters were estimated for the image by considering windows of size 4x4, 8x8, 16x16, 32x32 and 64x64. The mean square error ϵ_w was also computed. This procedure was done for different window size and different ISEED values. From all such different error values we have taken the maximum error for a particular window size. As mentioned earlier the errors also vary randomly with different ISEED values. The errors were then plotted against the window size. As expected, the error was minimum for window size 64x64. In Table 5.1, we present the error (maximum) for corresponding window size.

Table 5.1

| WINDOW | ERROR (Max.) |
|--------|-------------------------|
| 64 | 0.9098×10^{-2} |
| 32 | 0.2912×10^{-1} |
| 16 | 0.3556×10^{-1} |
| 8 | 0.9455×10^{-1} |
| 4 | 0.293829 |
| 2 | 0.352612 |

The error vs. window size plot as shown in Fig. 5.6, suggests that there may be an exponential relation between error and window size. This experiment was carried out for homogeneous images with different parameter sets. We have repeated the experiment by using different image sizes like 48x48 etc. In all such cases, we have found that error and window size are related by the equation

$$\epsilon_w = A w^{-B} \quad (5.8)$$

where,

ϵ_w = mean square error

w = window size

For the image with $\underline{q} = [0.12 \ 0.12]^T$ and $f = 1.1111$, if we estimate it as a 2 point problem then the eqn. (5.8) becomes

$$\epsilon_w = 1.456 w^{-1.297}$$

But estimating parameters for a 4 point neighbourhood for the same image gave

$$\epsilon_w = 1.46 w^{-1.949}$$

By changing either image size or the ISEED value it has been seen that A and B change. From simulation of different synthetic textures we have seen that A and B depend on i) number of parameters, ii) image size and iii) the ISEED value. As the ISEED value can be varied randomly hence A and B also vary in a random fashion. For this reason

we have given different plots for this error. vs. window size in Fig. 5.7 to show the variation of error with ISEED and window size. Apparently, it is impossible to give any mathematical expression for A and B in terms of the above mentioned variables. We have computed the error of such a scheme for two cases

- i) Taking $(\underline{e}, \underline{p})$ as the feature vector
- ii) Taking \underline{e} as the feature vector

From the plots it is seen that the additional information obtained by including \underline{p} in the feature vector does not change the error appreciably. We have also ^{computed} discrepancy in terms of the window size, where discrepancy is defined as follows.

If straight

line represents the actual segmenting line, then the hatched portion is defined to be the region of confidence as shown in Fig. 5.8

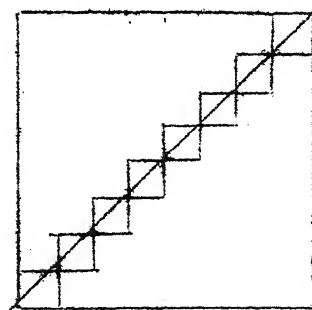


Fig. 5.8: Region of Confidence

Same concepts can be generalized to other segmenting curves. If the estimated segmenting line passes through this region we say that the segmentation procedure has converged

perfectly. But in actual practice, the segmenting line may deviated out of this region. We define discrepancy as the ratio of number of blocks outside the region to the total number of blocks. This discrepancy follows (Fig. 5.9) an exponential relationship with the window size and it is reduced by considering larger neighbourhood. One more thing to be noticed here is that the discrepancy-window size relation is independent of the segmenting curve. All these experiments mentioned above were done for various types of segmenting curves like st. line, circles, st. line with double slope etc. and all the results for them were consistent.

We conclude our study of texture segmentation by discussing some of the disadvantages of the scheme developed here.

For the purpose of elaboration, we consider the two-region problem again. Let us assume that the two parameters are (θ_1, ϕ_1) and (θ_2, ϕ_2) which have been used for generation. At the time of block estimation, we observe that estimate of θ_1 or θ_2 vary randomly about the actual θ_1 or θ_2 . Let us say $\hat{\theta}_1$ and $\hat{\theta}_2$ are the estimate of θ_1 and θ_2 .

Let,

$$\theta_{1L} \leq \hat{\theta}_1 \leq \theta_{1H}$$

$$\text{and, } \theta_{2L} \leq \hat{\theta}_2 \leq \theta_{2H}$$

where \underline{e}_{1L} and \underline{e}_{1H} are the lower value and upper value of the estimate of \underline{e}_1 respectively.

If we select \underline{e}_1 , \underline{e}_2 and the window size such that the following conditions are satisfied,

$$i) \quad | \underline{e}_{1L} - \underline{e}_{2L} | \gg 0$$

$$ii) \quad | \underline{e}_{1H} - \underline{e}_{2H} | \gg 0 \quad \text{and}$$

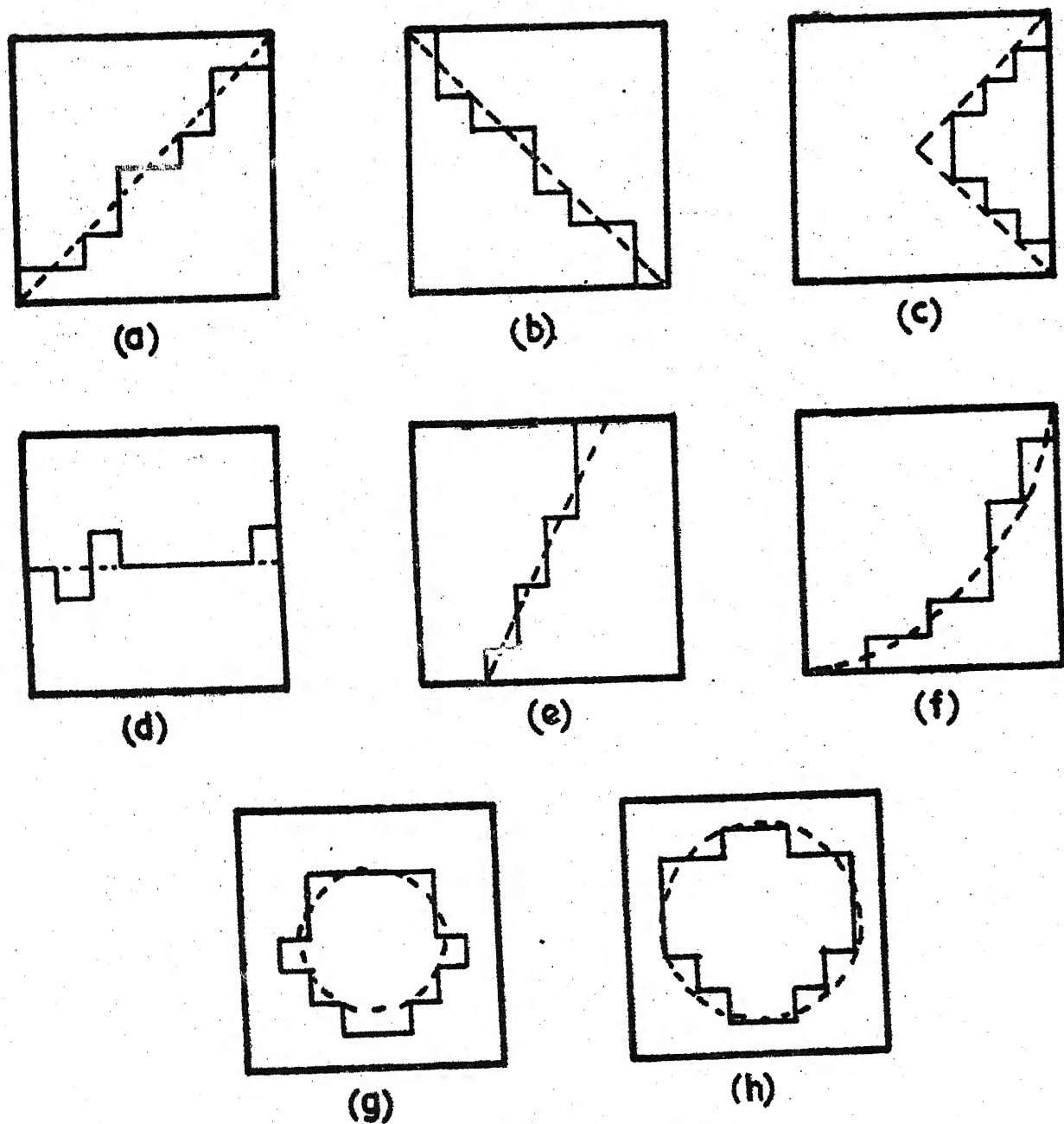
$$iii) \quad | \underline{e}_{1H} - \underline{e}_{2L} | \gg 0$$

where $| \underline{X} - \underline{Y} |$ means the absolute difference between the vectors \underline{X} and \underline{Y} , then the estimates of \underline{e}_1 and \underline{e}_2 are well separated, and the clustering scheme will work perfectly. But suppose, for a particular \underline{e}_1 and \underline{e}_2 , the conditions i) and ii) are satisfied but instead of iii) we have

$$\underline{e}_{1H} \cong \underline{e}_{2L}$$

then at the time of clustering, we may find intersections of such two clusters for some data points. Though we can assign these points to any of these clusters arbitrarily, if the number of such points is large, then clustering scheme may give erroneous segmenting line. That is to say that if the differences between the corresponding elements in the vectors \underline{e}_1 and \underline{e}_2 is not greater than a

threshold value this scheme may fail to detect the actual segmenting line. This is because of the limitations of the estimation rule. If for some θ_1 and θ_2 the scheme fails to detect the line properly we may say that statistically the texture is not a composite one and it may be treated as a homogeneous one.



— Estimated
 ---- Actual

FIG.5.5 THE ORIGINAL AND ESTIMATED VERSION OF DIFFERENT SEGMENTING CURVE FOR A 64x64 IMAGE.

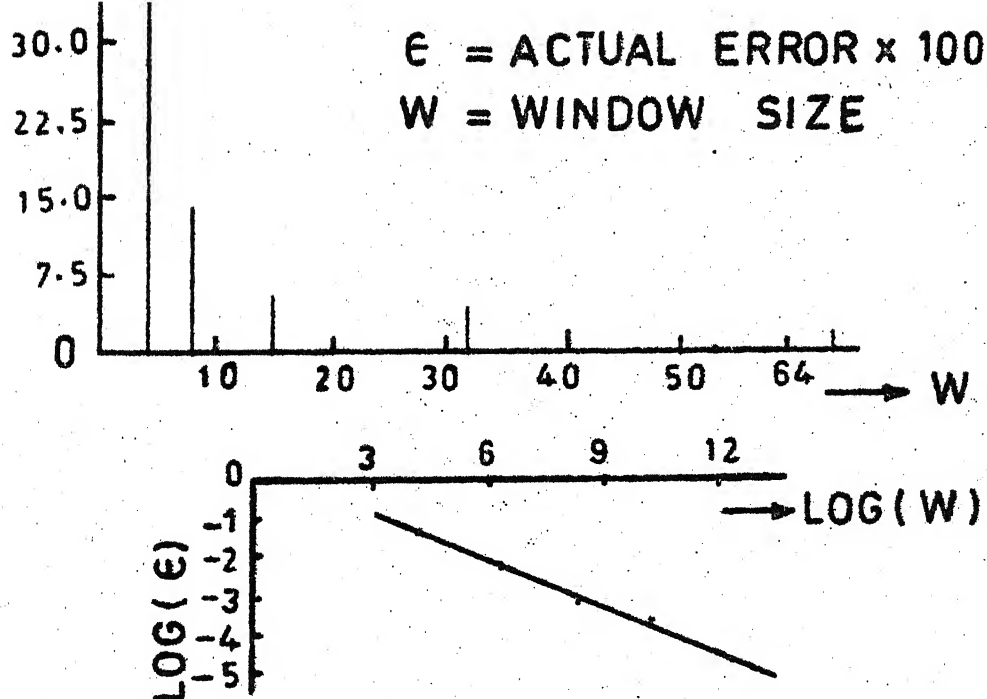


FIG.5.6(a) ERROR VERSUS WINDOW PLOT FOR A 64×64 SYNTHETIC TEXTURE (4 POINT ESTIMATION)

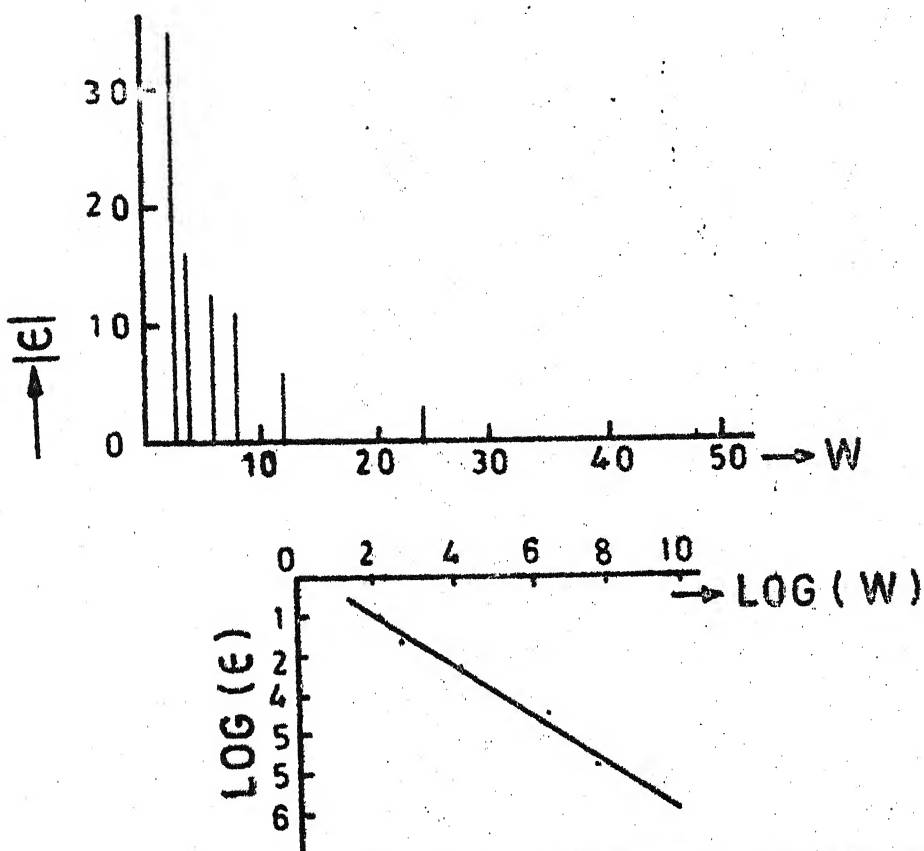


FIG.5.6(b) ERROR VERSUS WINDOW PLOT FO

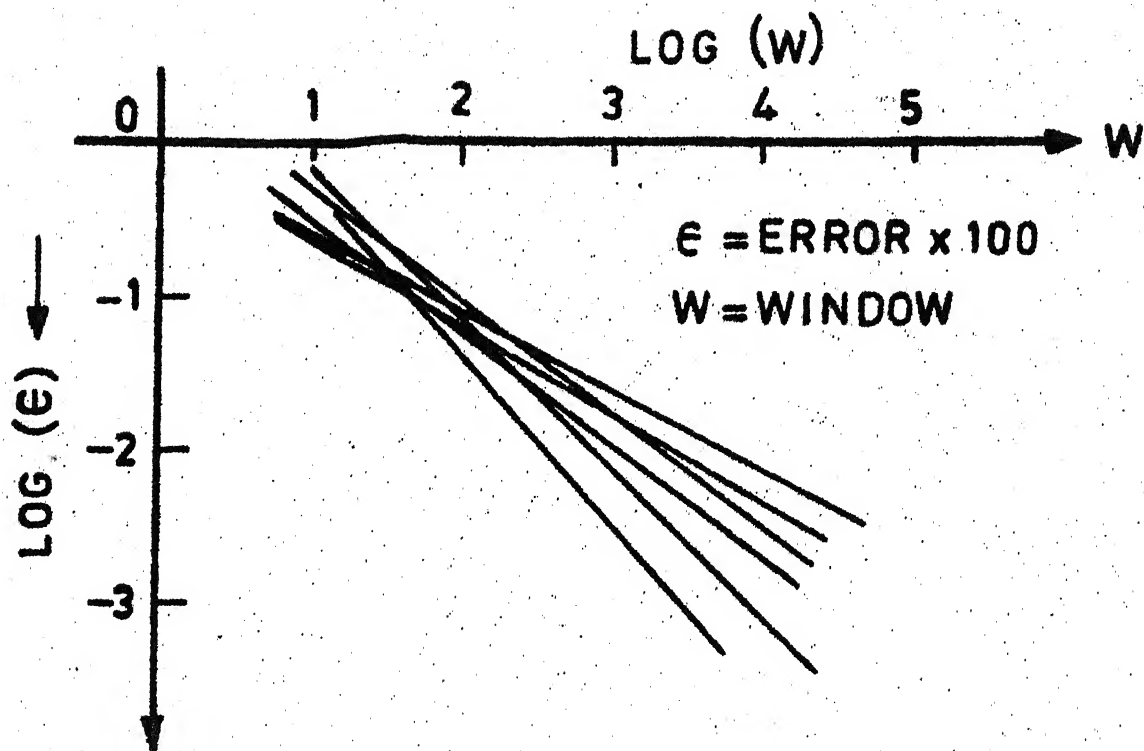


FIG. 5.7 THE ERROR VERSUS WINDOW FOR DIFFERENT ISEED VALUE.

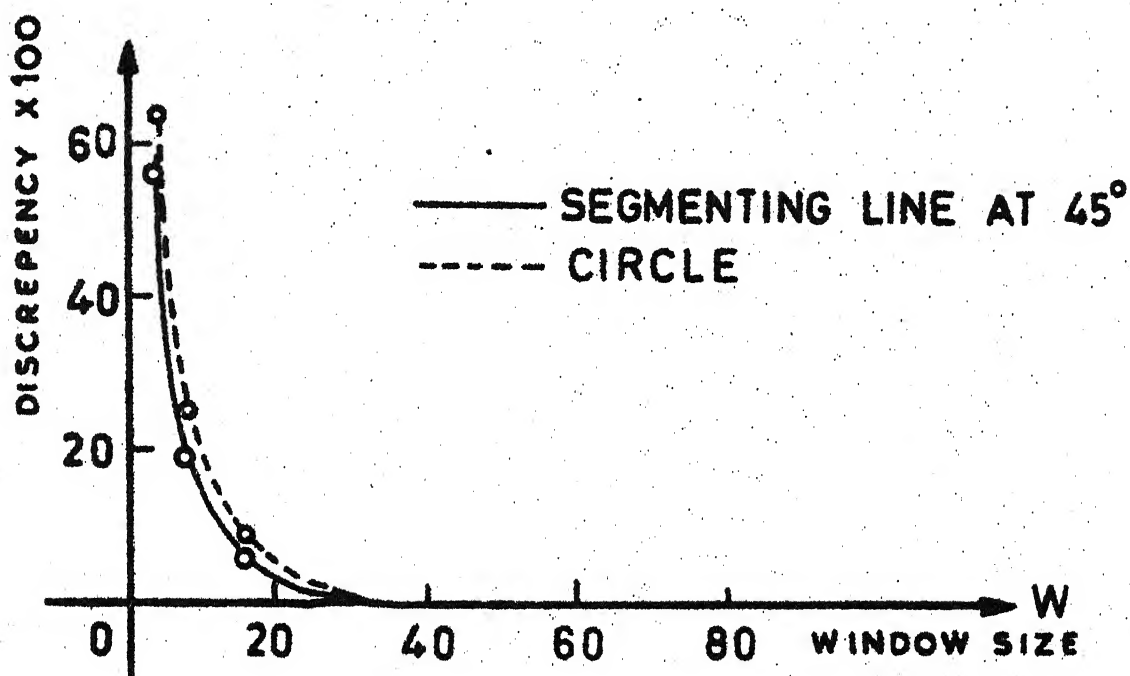


FIG. 5.9 DISCREPANCY VERSUS WINDOW FOR A 64×64 SYNTHETIC IMAGE.

CHAPTER 6

CONCLUSION

In our study of texture modelling, we have considered only the statistical modelling concept and the special type of statistical model, considered here, is known as 'random field' model. In the random field models, we have considered both causal and noncausal neighbourhoods. Procedures for implementation of such models have been given in details. We have cited examples of application of such models to texture segmentation and real image synthesis. A complete methodology has also been given for frequency domain characterization of textures. The application of 2D time series modelling to textural image has been studied for different Gaussian textures. We have also studied the applicability and limitations of pseudo random arrays in image modelling.

A new method for image segmentation for a composite image by SAR models is investigated. The area of image modelling is a vast in scope and the present study only considers one specific class of models. We have not specific class of models. We have not even touched upon the entire field of syntactic image modelling.

Some future areas of investigation are cyclostationary models and time series models other than linear Gaussian. Though linear non-Gaussian time series models are very useful for modelling of some real data, the time series models, having random parameters are very useful for real life image modelling. A large amount of research into these models is now being concentrated on the construction and application of computationally efficient algorithms to determine order and obtain estimates of the unknown parameters which have desirable statistical properties [40]. The development of 2D counterpart of the varying parameter time series model may show a new direction for image modelling.

REFERENCES

- [1] Weszka, J., Dyer, C., and Rosenfeld, A., 'A comparative study of texture measures for terrain classification', IEEE Trans. Syst., Man, Cybern. SMC-6 (1976), 269-285.
- [2] Haralick, R.M., Shanmugam, K., and Dinstein, I., 'Textural features for image classification', IEEE Trans. Syst., Man, Cybern. SMC-3(1973), 610-621.
- [3] Pavlidis, T. Structural pattern recognition, Springer - Verlag, New York, 1977.
- [4] Dungan, W., 'A terrain and cloud computer generation model', Comput. Gr. Image Process. 13(1979), 143-150.
- [5] Kashyap, R.L., Delp, E.J., and Mitchell, O.R., 'Image data compression using AR time series models', Pattern Recognition, vol. 11 (1979), 313-323.
- [6] Newman, W.M., and Sproull, R.F. Principles of interactive computer graphics, McGraw-Hill, Kogakusha, Tokyo, 1979.
- [7] Julesz, B., 'Visual pattern discrimination', IRE Trans. Inform. Theory 8,2 (1962) 84-92.
- [8] Ballard, D.H., and Brown, C.M., Computer vision, Prentice Hall, Englewood Cliffs, 1982.
- [9] Ahuja, N., and Schacter, B.J., 'Image Models', Computing Surveys, vol. 13, No. 4, December, 1981.
- [10] Kashyap, R.L., 'Random field models of images', Comput. Gr. Image Process. 12,3 (1980) 257-270.
- [11] Hunt, B.R., and Cannon, T.M., 'Nonstationary assumptions for Gaussian models of images', IEEE Trans. Syst., Man, Cybern. SMC-6 (1976), 876-882.
- [12] Hunt B.R., 'Bayesian methods in nonlinear image restoration', IEEE Trans. Comput. C-26 (1977) 219-229.
- [13] Hunt B.R., 'Nonstationary statistical image models (and their application to image data computation)', Comput. Gr. Image Process. 12,2 (1980), 173-186.

- [14] Trussel, H.J., and Kruger, R.P., 'Comments on 'nonstationary' assumptions for Gaussian models in images'', IEEE Trans. Syst. Man, Cybern. SMC-8 (1978), 579-582.
- [15] Nahi, N.E., and Jananshahi, M.H., 'Image boundary estimation', IEEE Trans. Comput. C-26, (1977), 772-781.
- [16] Jain, A.K., 'Partial differential equations and finite - difference methods in image processing, Part 1: Image representation', J. Optimiz. Theory Appln. 23(1977), 65-91.
- [17] Pratt, W.K., Faugeras, O.D., and Gagalowicz, A., 'Visual discrimination of stochastic texture fields', IEEE Trans. Syst. Man, Cybern. SMC-8(1978), 796-804.
- [18] Gagalowicz, A., 'Analysis of texture using a stochastic model', in Proc. 4th Int. Jr. Conf. Pattern Recognition, Nov. 1978, pp. 541-544.
- [19] Pratt, W.K., and Faugeras, D.O., 'Development and evaluation of stochastic-based visual texture features', in Proc. 4th Int. Jr. Conf. Pattern Recognition, Nov. 1978, pp. 545-548.
- [20] Whittle, P. 'On stationary processes in the plane', Biometrika 41(1954), 434-449.
- [21] Bartlett, M.S., 'Inference and stochastic processes', J.R. Stat. Soc. A 130 (1967) 457-477.
- [22] Besag, J., 'Spatial interaction and the statistical analysis of lattice system', J. Royal Stat. Soc. Ser. B 36,2(1974), 192-236.
- [23] Moran, P.A.P. and Besag, J., 'On the estimation and testing of spatial interaction in Gaussian lattice', Biometrika, vol. 62, no. 3(1975), 555-562.
- [24] Kashyap, R.L., 'Random field models on finite lattice for finite images', unpublished note.
- [25] Kashyap, R.L., 'Univariate and multivariate random field models for images', Comput. Gr. Image Process. 12(1980), 257-270.
- [26] Rosenfeld, A., and Troy, E., 'Visual texture analysis', in Conf. record for Symp. on feature extraction and selection in pattern recognition, Argonne, (1970), 115-124.

- [27] Haralick, R.M., 'Statistical and structural approaches to texture', in Proc. 4th Int. Jt. Conf. Pattern Recognition (1978), 45-69.
- [28] Box, J.E.P., and Jenkins, G.M. Time series analysis, Holden-Day, San Fransisco, 1976.
- [29] Jain, A.K., 'Advances in mathematical models for image processing', Proc. IEEE, vol. 69, no. 5(1981) 502-528.
- [30] Tou, J.T., 'Pictorial feature extraction and recognition via image modelling', Comput. Gr. Image Process., vol. 12(1980), 376-406.
- [31] MacWilliams, F.J. and Sloane, N.J.A., 'Pseudo-random sequences and arrays', Proc. IEEE, vol. 64(1976), 1715-1729.
- [32] Gill, A., 'Linear Sequential circuits', McGraw-Hill, New York(1966).
- [33] MacWilliams, F.J., and Sloane, N.J.A., 'The theory of error correcting codes,' North Holland, Amotrdam, 1977.
- [34] Golomb, S.W., Shift register sequences, Holden-Day, San-Fransisco, 1967.
- [35] Kashyap R.L., and Chellappa, R., 'Estimation and choice of neighbourhood in spatial interaction models of images', IEEE Trans. on Inf. Theory, vol. IT-29(1) 60-72(1983).
- [36] Woods, J.W., 'Markov image modelling', IEEE Trans. Automat. Contr., vol. AC-23(1978), 846-850.
- [37] Ord, K., 'Estimation methods for models of spatial interaction', J. Amer. Stat. Assoc., vol. 70(1975), 120-126.
- [38] Duda, R.O., and Hart, P.E., 'Pattern classification and scene analysis', Wiley, New York, 1973.
- [39] Tou, J.T., and Gonzalez, R.C., 'Pattern recognition principles,' Addison-wesley, Reading, 1974.
- [40] Nicholls, D.F., and Quinn, B.G., Random Coefficient Autoregressive Models: An Introduction, Springer-Verlag, New York.

A 87432

EE-1983-M-BAN-STU